i.e.



1982 NO. 2
HK$6.00

*6502←→6809製作

*AM立體聲最新動向

*ATARICUP：最勁的跑車遊戲　全部LISTINGS落齊

勁 GAMES!

綜合性“6”字頭雜誌

編後REM

• “6”字頭包括了6502, 6800, 6809, 68000等
故此包括了很多機種，但相信最熱門的仍
然是APPLE, ATARI及VIC－20。

• 但還有TRS-80C, SUPER BOARD, AIM 65及ATOM
之類的亦不小。

• 本刊總部將所有6502“HACKER”（機迷）通通召集
一起，暫停其它稿件，集中所有ENERGY, 籌備這一
本6502專刊。

• 要保証所有程式都要夠勁！

• 本刊内容經多次研究，大家認爲質素不差，
銷路有把握。

• 唯一是要找一批高手「壓陣」，終於先後找了ATARI
的BAN SZETO, APPLE的“人腦”。其中以BAN
SZETO的“大賽車”已一早可以同鬼佬拗手瓜。
有了他作帶頭，相信可以號召其他ATARI作
者加入本刊陣營，令IE6502變爲超級電腦雜誌。

歡迎介紹，歡迎批評，歡迎訂閱。

Interface Electronics 6502 for
6502 6809 6800 68000
6............cpu
SEP. 1982
2

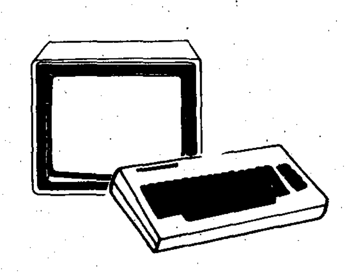
目錄

GAMES



製作專欄

新機介紹

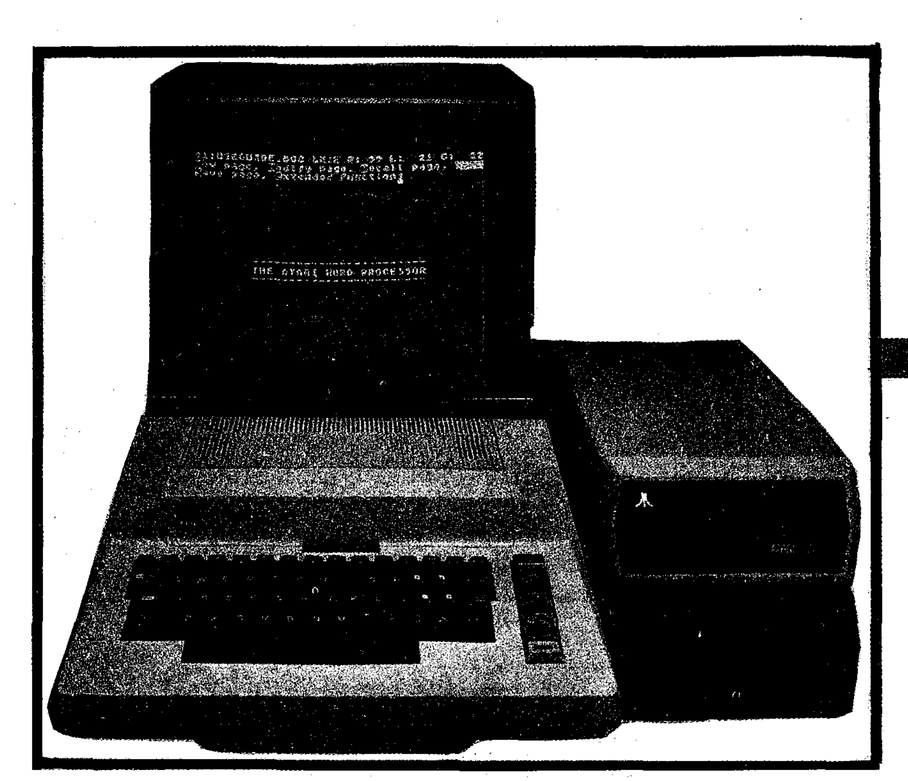


主　編：譚　言
李少龍
發　行：同德書報社
廣　告：鄧耀發
承　印：堅利印刷有限公司
執行編輯：冼金清
吳偉聰
通訊地址：九龍新蒲崗行慶街行慶大厦
六樓B3室 TEL-237644
制作助理：廖端儀
出　版：INTERFACE ELECTRONIC PUBLISHERS

新產品介紹

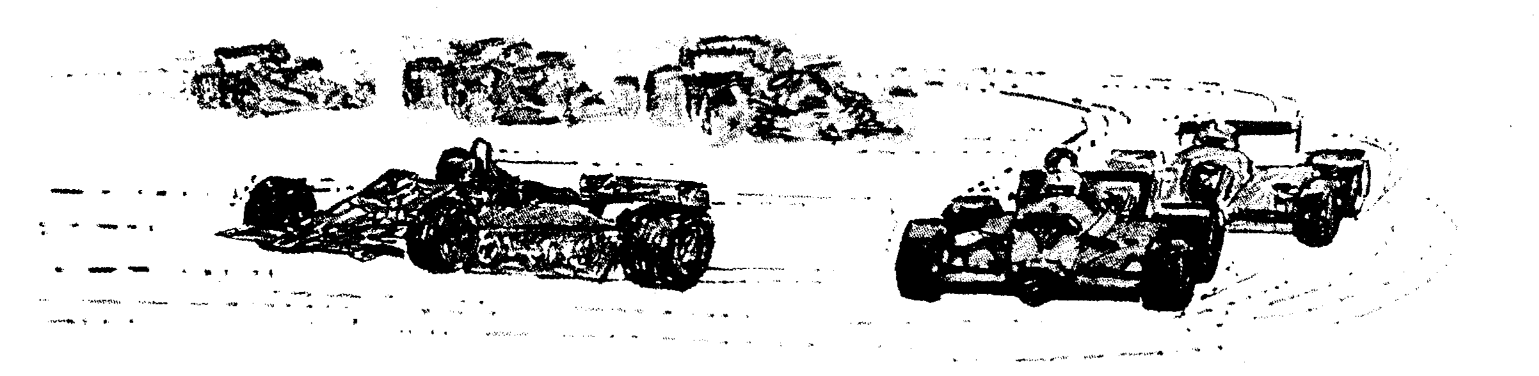
電子技術

其他



ATARI

瘋狂大賽車



筆者最近編寫了一個很刺激的賽車遊戲。爲了使讀者能夠和筆者分享這個程式的樂趣，所以把這個程式列了出來。

整個程式可以分成三部份。第一部份是一個BASIC程式（HEX INPUT），用來把第二部份的機械碼輸進電腦裏的。只要把HEX INPUT的程式輸入電腦，跟著，打入RUN。程式便會問程序員的開始；這樣程序員必須根據各機械語言的開始地址，輸入適當機械碼。如果輸入時發覺錯誤，可以從新RUN HEX INPUT這個程式，再輸入機械碼。

1分53秒正

當所有機械碼輸入完畢後，程序員便應該打入GOTO 5000。這樣HEX INPUT的責任便告一個段落。用者可以自行把它SAVE過來。跟著便到第三部份。第三部份便是遊戲的主要部份。當輸入完畢後，便可打入RUN，遊戲便會隨卽開始。

玩者必須以最快速的時間到達終點。不過，如果在駕駛時撞到其他的車，便會失去5秒鐘的時間。筆者曾經創下了1分53秒的紀錄，如果讀者有信心的話，不妨向筆者挑戰。

HEX INPUT PROGRAM

```
5 GOSUB 500
10 PRINT "STARTING ADDRESS ";
20 INPUT STADD$
30 IN$=STADD$:GOSUB 1000
40 N=OUT
50 GOSUB 2000
60 N=N+1:GOTO 50
500 DIM IN$(4),STADD$(4),OUT$(4),HEX$(2)
510 CLOSE #1
520 OPEN #1,4,0,"K:"
599 RETURN
1000 REM HEX TO DEC
1005 OUT=0
1010 FOR L=1 TO LEN(IN$)
1020 DEC=ASC(IN$(L,L))-48
1030 IF DEC>9 THEN DEC=DEC-7
1040 OUT=OUT*16+DEC
1050 NEXT L
1060 RETURN
2000 IF N=OUT THEN ? STADD$;" ";:GOTO 2020
2010 IF N-INT(N/8)*8=0 THEN IN=N:GOSUB 3000:? :? OUT$;" ";
2020 GOSUB 4000
2030 HEX$(1,1)=CHR$(Y)
2040 GOSUB 4000
2050 HEX$(2,2)=CHR$(Y)
2060 IN$=HEX$:GOSUB 1000
2070 POKE N,OUT
2075 ? " ";
2080 RETURN
3000 REM DEC TO HEX
3005 OUT$=""
3010 FOR L=4 TO 1 STEP -1
3020 Y=IN
3030 IN=INT(IN/16)
3040 Y=Y-IN*16
3050 OUT$(L,L)=CHR$(Y+48+(Y>9)*7)
3060 NEXT L
3070 RETURN
4000 GET #1,Y:IF Y>ASC("F") OR Y<ASC("0") OR Y>ASC("9")
     AND Y<ASC("A") THEN 4000
4010 ? CHR$(Y);
4020 RETURN
5000 CLOSE #1
5010 OPEN #1,8,0,"D:ATARICUP.MAC"
5020 FOR N=1536 TO 1791
5030 PUT #1,PEEK(N)
5040 NEXT N
5050 FOR N=27904 TO 29439
5060 PUT #1,PEEK(N)
5070 NEXT N
5080 FOR N=30720 TO 31775
5090 PUT #1,PEEK(N)
5100 NEXT N
5110 CLOSE #1
5120 OPEN #1,8,0,"D:TOPREC.DAT"
5130 FOR N=1 TO 20
5140 ? #1;150
5150 ? #1;"NOBODY      "
5160 ? #1;"??-??-19??"
5170 NEXT N
5180 CLOSE #1
9999 END
```

OBJECT FILES for ATARICUP.BAS

PART1.OBJ : 7000-72FF

```
7000 A9 98 8D 00 D0 8D 2A 06 A9 6D 8D 2E 06 8D 04 D0
7010 A9 D1 8D 05 D0 8D 2F 06 A9 9F 8D 06 D0 A9 D2 85
7020 CB A9 74 85 CC A0 20 B9 00 06 91 CB 88 D0 F8 A9
7030 00 85 CB A9 73 85 CC A2 10 A9 0F A0 08 91 CB 88
7040 D0 FB A5 CB 18 69 10 85 CB CA D0 ED A9 14 85 CB
7050 A9 73 85 CC A2 04 A0 14 B1 CB 18 69 30 91 CB 88
7060 D0 F6 A5 CB 18 69 40 85 CB CA D0 EA A9 00 8D 35
7070 06 8D 36 06 8D 37 06 A9 D2 8D 30 06 A9 88 8D 5D
7080 06 A9 13 8D 39 06 A9 00 8D 38 06 8D 26 06 8D 27
7090 06 8D 28 06 8D 29 06 8D 22 06 8D 3A 06 8D 3B 06
70A0 8D 3C 06 8D 34 06 8D 21 06 8D 31 06 8D 32 06 8D
70B0 33 06 A9 2E 8D 01 D2 A9 60 8D 00 D2 A9 00 8D 5C
70C0 06 8D 5B 06 8D 5A 06 8D 55 06 8D 56 06 8D 57 06
70D0 8D 58 06 8D 59 06 8D 1E D0 60 AD 22 06 18 6D 26
70E0 06 8D 26 06 C9 20 B0 08 A2 00 A0 08 20 F8 71 60
70F0 48 29 1F 8D 26 06 68 4A 4A 4A 4A 4A 49 FF 38 69

7100 00 A8 AD 00 74 8D 3F 06 AD FF 73 8D 40 06 AD FE
7110 73 8D 41 06 AD FD 73 8D 42 06 B9 00 73 8D 00 74
7120 88 A2 FF B9 00 73 9D 00 73 CA 88 D0 F6 AD 3F 06
7130 9D 00 73 CA F0 18 AD 40 06 9D 00 73 CA F0 0F AD
7140 41 06 9D 00 73 CA F0 06 AD 42 06 9D 00 73 AD 3A
7150 06 D0 3B AD 3B 06 D0 6B AD 0A D2 C9 F0 90 12 AD
7160 0A D2 C9 F0 B0 0C AD 0A D2 C9 15 B0 04 C9 15 90
7170 0F 60 A9 01 8D 3A 06 AD 0A D2 29 07 8D 3D 06 60
7180 A9 01 8D 3B 06 AD 0A D2 29 F8 8D 3D 06 60 AD 2E
7190 06 C9 8C B0 28 EE 3C 06 AD 3C 06 C9 40 B0 01 60
71A0 A9 00 8D 3C 06 EE 2E 06 AD 2E 06 8D 04 D0 CE 2F
71B0 06 AD 2F 06 8D 05 D0 CE 3D 06 F0 01 60 A9 00 8D
71C0 3A 06 60 EE 3C 06 AD 3C 06 C9 40 B0 01 60 A9 00
71D0 8D 3C 06 AD 2F 06 C9 D1 B0 18 EE 2F 06 AD 2F 06
71E0 8D 05 D0 CE 2E 06 AD 2E 06 8D 04 D0 CE 3D 06 F0
71F0 01 60 A9 00 8D 3B 06 60 CA D0 FD 88 D0 FA 60 AD
7200 78 02 29 01 F0 03 4C 22 72 AD 22 06 C9 7F B0 3B
7210 AD 21 06 18 69 80 8D 21 06 AD 22 06 69 00 8D 22
7220 06 60 A9 06 8D 3E 06 AD 22 06 F0 1F AD 78 02 29
7230 02 D0 05 A9 E0 8D 3E 06 AD 21 06 38 ED 3E 06 8D
7240 21 06 AD 22 06 E9 00 8D 22 06 60 60 AD 22 06 4A
7250 8D 3F 06 AD 22 06 4A 4A 18 6D 3F 06 8D 3F 06 A9
7260 D2 38 ED 3F 06 CD 30 06 F0 04 B0 0C 90 23 A2 50
7270 A0 01 20 F8 71 4C AC 72 EE 30 06 AD 30 06 85 CB
7280 A9 74 85 CC A0 20 B9 00 06 91 CB 88 10 F8 4C AC
7290 72 CE 30 06 AD 30 06 85 CB A9 74 85 CC A0 21 A9
72A0 00 91 CB 88 B9 00 06 91 CB 88 D0 F8 EE 34 06 AD
72B0 34 06 C9 02 B0 01 60 A9 00 8D 34 06 AD 78 02 29
72C0 04 F0 08 AD 78 02 29 08 F0 0B 60 CE 2A 06 AD 2A
72D0 06 8D 00 D0 60 EE 2A 06 AD 2A 06 8D 00 D0 60 00
72E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
72F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```





PART2.OBJ : 7800-7C1F

```
7800 00 A9 00 8D 53 06 20 23 78 20 71 7A A9 01 8D 53
7810 06 20 23 78 20 71 7A A9 02 8D 53 06 20 23 78 20
7820 71 7A 60 AE 53 06 AD 53 06 C9 00 F0 12 C9 01 F0
7830 07 A9 77 85 CC 4C 43 78 A9 76 85 CC 4C 43 78 A9
7840 75 85 CC BD 35 06 D0 03 4C A8 79 BD 27 06 38 FD
7850 23 06 18 6D 22 06 30 03 4C 6E 79 49 FF 18 69 01
7860 C9 20 B0 0B 49 FF 18 69 01 9D 27 06 4C A0 79 8D
7870 3F 06 29 1F 49 FF 18 69 01 AE 53 06 9D 27 06 AD
7880 3F 06 4A 4A 4A 4A 4A 8D 3F 06 BD 31 06 38 ED 3F
7890 06 B0 03 FE 50 06 9D 31 06 BD 50 06 C9 02 90 0A
78A0 BD 31 06 C9 B0 90 46 4C B8 78 BD 50 06 C9 01 B0
78B0 07 BD 31 06 C9 B0 B0 35 BD 50 06 F0 33 C9 01 F0
78C0 37 BD 31 06 C9 E2 90 28 BD 31 06 18 69 24 A8 A2
78D0 20 A9 00 85 CB A9 00 91 CB 88 91 CB 88 91 CB 88
78E0 91 CB 88 BD 00 06 91 CB CA 88 D0 F7 60 4C 56 79
78F0 A2 00 A0 01 20 F8 71 60 BD 31 06 C9 E0 B0 32 A9
7900 00 85 CB BD 31 06 18 69 24 A8 A9 00 91 CB 88 91
7910 CB 88 91 CB 88 91 CB 88 A2 20 BD 00 06 91 CB 88
7920 CA D0 F7 A9 00 91 CB 88 91 CB 88 91 CB 88 91 CB
7930 60 A9 00 85 CB BD 31 06 38 E9 04 A8 A9 00 91 CB
7940 C8 91 CB C8 91 CB C8 91 CB CB A2 01 BD 00 06 91
7950 CB E8 C8 D0 F7 60 A9 00 9D 35 06 9D 27 06 20 62
7960 79 60 A0 00 A9 00 85 CB 91 CB 88 D0 FB 60 C9 20
7970 B0 06 9D 27 06 4C A0 79 8D 3F 06 29 1F 9D 27 06
7980 AD 3F 06 4A 4A 4A 4A 4A 8D 3F 06 BD 31 06 18 6D
7990 3F 06 9D 31 06 B0 03 4C 99 78 DE 50 06 4C 99 78
79A0 A2 01 A0 02 20 F8 71 60 AE 53 06 E8 8A 29 03 AA
79B0 BD 50 06 C9 02 90 14 E8 8A 29 03 AA BD 50 06 C9
79C0 02 90 08 A2 28 A0 02 20 F8 71 60 AE 53 06 A9 01
79D0 9D 35 06 A9 C0 9D 31 06 A9 02 9D 50 06 AD 0A D2
79E0 29 F0 C9 60 F0 F7 C9 80 F0 F3 C9 A0 F0 EF C9 C0
79F0 F0 EB C9 03 F0 E7 9D C1 02 AD 0A D2 29 03 18 69
7A00 07 8D 3F 06 BD C1 02 18 6D 3F 06 9D C1 02 AD 0A
7A10 D2 29 0F 18 69 03 9D 4D 06 AD 0A D2 29 0F 18 69
7A20 45 9D 23 06 AD 2F 06 38 E9 12 8D 3F 06 AD 2E 06
7A30 18 69 06 8D 40 06 AD 0A D2 CD 3F 06 B0 F8 CD 40
7A40 06 90 F3 9D 2B 06 9D 01 D0 AD 0A D2 C9 80 B0 0B
7A50 90 14 A9 00 9D 47 06 9D 4A 06 60 A9 00 9D 4A 06
7A60 A9 01 9D 47 06 60 A9 00 9D 47 06 A9 01 9D 4A 06
7A70 60 AE 53 06 FE 44 06 BD 44 06 DD 4D 06 B0 01 60
7A80 A9 00 9D 44 06 BD 47 06 D0 1A BD 4A 06 D0 3D BD
7A90 2B 06 18 69 14 38 ED 2F 06 B0 46 BD 2B 06 38 E9
7AA0 0A 90 44 60 BD 2B 06 38 E9 06 CD 2E 06 90 0A DE
7AB0 2B 06 BD 2B 06 9D 01 D0 60 BD 47 06 8D 3F 06 BD
7AC0 4A 06 9D 47 06 AD 3F 06 9D 4A 06 60 BD 2B 06 18
7AD0 69 12 CD 2F 06 B0 E2 FE 2B 06 BD 2B 06 9D 01 D0
7AE0 60 A9 01 9D 47 06 60 A9 01 9D 4A 06 60 00 00 00
7AF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7B00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7B10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7B20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7B30 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7B40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7B50 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7B60 A9 3E 8D 2F 02 A9 03 8D 1D D0 A9 70 8D 07 D4 A9
7B70 11 8D 6F 02 A9 55 8D 0C D0 A9 3A 8D C0 02 A9 0A
7B80 8D C7 02 A9 01 8D 08 D0 8D 09 D0 8D 0A D0 8D 0B
7B90 D0 A9 00 8D 00 D0 8D 01 D0 8D 02 D0 8D 03 D0 8D
7BA0 04 D0 8D 05 D0 8D 06 D0 8D 07 D0 A9 73 85 CC A9
7BB0 00 85 CB A0 00 91 CB 88 D0 FB A5 CC 18 69 01 C9
7BC0 78 F0 07 85 CC A9 00 4C B5 7B 60 60 00 00 00 00
7BD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7BE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7BF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7C00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
7C10 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

```
x 7B60 A9 3E                      LDA    #$3E
  7B62 8D 2F 02                   STA    $022F
  7B65 A9 03                      LDA    #$03
```

PART31.OBJ : 0600-06CF

```
0600 00 3C 3C 3C 18 5A 5A 5A 7E 5A 5A 5A 18 3C 3C 3C
0610 3C 3C 3C 3C 3C FF FF DB FF DB DB DB 18 3C 7E 7E
0620 7E 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0630 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0640 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0650 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0660 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0670 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0680 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0690 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
06A0 68 20 60 7B 20 00 70 20 01 78 20 DA 70 20 00 6E
06B0 20 FF 71 20 4C 72 20 12 6F 20 1A 6E 20 56 6F AD
06C0 5C 06 D0 03 4C A7 06 60 00 00 00 00 00 00 00 00
```

PART32.OBJ : 6D00-6FFF

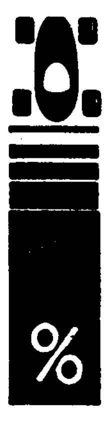
```
6D00 AD 5A 06 18 69 18 8D 5A 06 AD 55 06 69 01 C9 0A
6D10 B0 06 8D 55 06 4C 5D 6D E9 0A 8D 55 06 EE 56 06
6D20 AD 56 06 C9 0A B0 03 4C 5D 6D E9 0A 8D 56 06 EE
6D30 57 06 AD 57 06 C9 0A B0 03 4C 5D 6D E9 0A 8D 57
6D40 06 EE 58 06 AD 58 06 C9 06 B0 03 4C 5D 6D E9 06
6D50 8D 58 06 EE 59 06 AD 59 06 C9 03 B0 30 20 9B 6D
6D60 A0 28 AD 59 06 18 69 10 91 CB C8 C8 AD 58 06 69
6D70 10 91 CB C8 AD 57 06 69 10 91 CB C8 C8 AD 56 06
6D80 69 10 91 CB C8 AD 55 06 69 10 91 CB 60 A9 01 8D
6D90 5E 06 8D 5C 06 A9 00 8D 5F 06 60 AD 30 02 85 CB
6DA0 AD 31 02 85 CC A0 04 B1 CB 8D 3F 06 C8 B1 CB 8D
6DB0 40 06 AD 3F 06 85 CB AD 40 06 85 CC 60 00 00 00
6DC0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6DD0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6DE0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6DF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
6E00 AD 22 06 CD 43 06 D0 01 60 8D 43 06 4A 8D 3F 06
6E10 A9 60 38 ED 3F 06 8D 00 D2 60 18 AD 08 D0 29 01
6E20 F0 15 20 94 6E AD 2E 06 18 69 06 8D 2A 06 8D 00
6E30 D0 A9 00 8D 1E D0 60 AD 09 D0 29 01 F0 15 20 94
6E40 6E AD 2F 06 38 E9 14 8D 2A 06 8D 00 D0 A9 00 8D
6E50 1E D0 60 AD 0C D0 D0 01 60 20 94 6E AD 0C D0 29
6E60 02 F0 09 A2 00 A9 75 85 CC 4C 82 6E AD 0C D0 29
6E70 04 F0 09 A2 01 A9 76 85 CC 4C 82 6E A2 02 A9 77
6E80 85 CC A9 00 9D 35 06 85 CB A0 00 91 CB 88 D0 FB
6E90 8D 1E D0 60 A9 8F 8D 01 D2 A9 0F 8D 54 06 AD 2A
6EA0 06 38 E9 01 8D 00 D0 A2 00 A0 20 20 F8 71 AD 54
6EB0 06 4A 8D 3F 06 A9 3F 38 ED 3F 06 8D 00 D2 AD 2A
6EC0 06 18 69 01 8D 00 D0 A2 00 A0 20 20 F8 71 A9 80
6ED0 18 6D 54 06 8D 01 D2 CE 54 06 D0 C2 A9 2E 8D 01
6EE0 D2 AD 57 06 18 69 05 C9 0A B0 04 8D 57 06 60 E9
6EF0 0A 8D 57 06 EE 58 06 AD 58 06 C9 06 B0 01 60 E9
6F00 06 8D 58 06 EE 59 06 AD 59 06 C9 03 B0 01 60 4C
6F10 8D 6D 20 00 6D 20 19 6F 60 AD 22 06 85 D4 4A 18
6F20 65 D4 85 D4 A9 00 85 D5 20 AA D9 20 E6 D8 20 9B
6F30 6D A2 00 A0 8F A9 00 91 CB 88 91 CB 88 88 BD 80
6F40 05 38 E9 20 91 CB 29 80 D0 05 E8 C8 4C 3E 6F B1
6F50 CB 29 7F 91 CB 60 AD 38 06 38 ED 22 06 8D 38 06
6F60 AD 5D 06 E9 00 8D 5D 06 90 03 4C 95 6F AD 39 06
6F70 C9 02 90 78 AD 39 06 F0 06 CE 39 06 4C 95 6F A9
6F80 01 8D 5C 06 8D 5F 06 A9 00 8D 5E 06 8D 38 06 8D
6F90 5D 06 8D 39 06 AD 5D 06 85 D4 AD 39 06 85 D5 20
6FA0 AA D9 20 E6 D8 20 9B 6D A0 F0 20 B9 6F AD 5B 06
6FB0 D0 01 60 A0 71 20 B9 6F 60 8C 3F 06 A9 00 91 CB
6FC0 C8 91 CB C8 91 CB C8 91 CB C8 91 CB C8 91 CB A2
6FD0 00 AC 3F 06 BD 80 05 38 E9 20 91 CB 29 80 D0 05
6FE0 E8 C8 4C D4 6F B1 CB 29 7F 91 CB 60 A9 01 8D 5B
6FF0 06 4C 74 6F 00 00 00 00 00 00 00 00 00 00 00 00
```

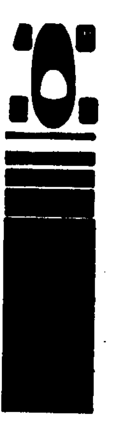
BANCUP.BAS by BAN SZETO

```
10 GOSUB 1000
130 GOSUB 4000
132 COUNT=A0
135 IF PEEK(STARTKEY)=6 THEN GOSUB 3000:GOTO 181
140 IF PEEK(STARTKEY)=5 THEN 159
155 POKE 77,A0
157 COUNT=COUNT+A1:IF COUNT<300 THEN 135
159 COUNT=A0:GOSUB 4500
160 POKE 764,255
161 IF PEEK(STARTKEY)=6 THEN GOSUB 3000:GOTO 181
163 IF PEEK(STARTKEY)=5 THEN 130
165 POKE 77,A0
170 COUNT=COUNT+1:IF COUNT<300 THEN 161
175 GOTO 130
181 IF PEEK(1630)=A1 AND PEEK(1631)=A0 THEN GOSUB 6000
182 IF PEEK(1630)=A0 AND PEEK(1631)=A1 THEN GOSUB 7000
185 GOSUB 2000
190 IF REC<RECORDMAX THEN GOSUB 5000
200 IF REC<TOP(20) THEN GOSUB 5800
230 GOTO 130
1000 A0=0:A1=1:A2=2:A4=4:A5=10
1008 OPEN #A1,A4,A0,"K:"
1011 GRAPHICS 17
1012 POKE 16,64
```

```
1013 DIM TODAY$(A5),TIME$(7),INNAME$(12),
     KPER$(12),DAT$(A5),T1$(A1),T2$(A1)
1020 DIM RECORD(A5)
1030 FOR N=A1 TO A5
1040 RECORD(N)=180:REM 3:00.00
1050 NEXT N
1055 DIM TOP(20),KEEPER$(240),DATE$(200)
1056 FOR N=A1 TO 20:TOP(N)=150:NEXT N
1057 FOR N=A1 TO 240:KEEPER$(N,N)=" ":NEXT N
1058 FOR N=A1 TO 200:DATE$(N,N)=" ":NEXT N
1060 RECORDMAX=180
1061 UPDATED=A0
1063 DIM TEMP$(12)
1065 DIM NAME$(120)
1066 FOR N=1 TO 120
1067 NAME$(N)=" "
1068 NEXT N
1070 STARTKEY=53279
1080 GOSUB 1500
1084 GOSUB 1650
1087 GOSUB 1200
1090 RETURN
1200 TODAY$="00-00-1900"
1210 GRAPHICS 0
1220 SETCOLOR 2,12,0
1230 POSITION 4,4:PRINT "SIGN ON"
1240 POSITION 12,8:PRINT "YEAR  : 1900"
1250 POSITION 12,10:PRINT "MONTH : 00"
1260 POSITION 12,12:PRINT "DAY   : 00"
1270 POSITION 21,8:PRINT "→";
1280 POKE 764,255
1290 GET #1,Y:IF Y>57 OR Y<48 THEN 1290
1295 PRINT CHR$(Y);
1300 TODAY$(9,9)=STR$(Y-48)
1310 GET #1,Y:IF Y>57 OR Y<48 THEN 1310
1315 PRINT CHR$(Y);
1320 TODAY$(10,10)=STR$(Y-48)
1330 POSITION 19,10:PRINT "→";
1335 POKE 764,255
1340 GET #1,Y:IF Y>57 OR Y<48 THEN 1340
1345 PRINT CHR$(Y);
1350 TODAY$(1,1)=STR$(Y-48):T1$=CHR$(Y)
1360 GET #1,Y:IF Y>57 OR Y<48 THEN 1360
1365 PRINT CHR$(Y);
1370 TODAY$(2,2)=STR$(Y-48)
1380 IF VAL(TODAY$(1,2))>12 THEN POSITION 20,
     10:PRINT "00":GOTO 1330
1390 POSITION 19,12:PRINT "→";
1395 POKE 764,255
1400 GET #1,Y:IF Y>57 OR Y<48 THEN 1400
1405 PRINT CHR$(Y);
1410 TODAY$(4,4)=STR$(Y-48):T2$=CHR$(Y)
1420 GET #1,Y:IF Y>57 OR Y<48 THEN 1420
1425 PRINT CHR$(Y);
1430 TODAY$(5,5)=STR$(Y-48)
1440 IF VAL(TODAY$(4,5))>31 THEN POSITION 20,
     12:PRINT "00":GOTO 1390
1450 POSITION 4,20:PRINT "ANY CHANGE? (Y/N)";
1460 GET #1,Y
1470 IF Y=ASC("Y") THEN 1240
1480 TODAY$(1,1)=T1$:TODAY$(4,4)=T2$
1496 RETURN
1500 IF PEEK(1537)=60 AND PEEK(31313)=20 THEN RETURN
1501 GOSUB 9000
1509 OPEN #2,4,0,"D1:ATARICUP.MAC"
1510 FOR N=1536 TO 1791
1520 GET #A2,Y
1530 POKE N,Y
1540 NEXT N
1550 FOR N=27904 TO 29439
1560 GET #A2,Y
1570 POKE N,Y
1580 NEXT N
1590 FOR N=30720 TO 31775
1600 GET #A2,Y
1610 POKE N,Y
1620 NEXT N
1630 CLOSE #A2
1640 RETURN
1650 POKE 764,12
1690 OPEN #A2,A4,A0,"D1:TOPREC.DAT"
1700 FOR N=A1 TO 20
1710 INPUT #2;TOP,KPER$,DAT$
1720 TOP(N)=TOP
1730 KEEPER$((N-A1)*12+A1,N*12)=KPER$
1740 DATE$((N-A1)*A5+A1,N*A5)=DAT$
1750 NEXT N
1760 CLOSE #A2
1770 RETURN
2000 TIMECT=1621
2010 REC=60*PEEK(TIMECT+A4)+A5*PEEK(TIMECT+3)
     +PEEK(TIMECT+A2)+PEEK(TIMECT+A1)/A5
+PEEK(TIMECT)/100
2020 RETURN
3000 FOR N=53248 TO 53255:POKE N,A0:NEXT N
3020 GRAPHICS 17
3021 POKE 559,62
3022 SETCOLOR A1,14,8
3023 SETCOLOR A2,5,8
3024 SETCOLOR A3,12,8
3025 SETCOLOR A0,11,6
3030 POSITION A0,A0:PRINT #6;"time:"
3035 POSITION A0,A2:PRINT #6;"0:00.00";
3040 POSITION A0,5:PRINT #6;"speed:"
3045 POSITION A0,7:PRINT #6;"0"
3050 POSITION 3,8:PRINT #6;"KM/H"
3060 POSITION A0,A5:PRINT #6;"distant"
3065 POSITION A0,12:PRINT #6;"5000"
3070 POSITION 6,12:PRINT #6;"m";
3080 POSITION A5,6:PRINT #6;"READY"
3081 CT=A0:POSITION A0,14:? #6;"OVERALL":? #6;"FASTEST":? #6
3082 MIN=INT(TOP(1)/60):SEC=TOP(1)-MIN*60:? #6;MIN;":";SEC:? #6
3083 ? #6;"by :":? #6
3084 CT=CT+A1:IF KEEPER$(CT,CT)=" " OR CT=8 THEN 3086
3085 ? #6;KEEPER$(CT,CT);:GOTO 3084
3086 ? #6
3087 CT=CT+A1:IF KEEPER$(CT,CT)=" " OR CT=12 THEN 3090
3088 ? #6;KEEPER$(CT,CT);:GOTO 3087
3090 FOR N=A0 TO 7 STEP 0.05
3100 SOUND A0,96,2,N
3102 IF N=6 THEN POSITION A5,6:PRINT #6;"      "
3105 NEXT N
3110 POSITION A5,6:PRINT #6;" GO! "
3115 FOR N=8 TO 14 STEP 0.05
3118 SOUND A0,96,A2,N
3119 IF N=13 THEN POSITION A5,6:PRINT #6;"      "
3121 NEXT N
3124 POSITION A5,6:PRINT #6;"      "
3130 X=USR(1696)
3140 RETURN
4000 FOR N=53248 TO 53255:POKE N,A0:NEXT N
4020 GRAPHICS 17
4030 POKE 559,62
4032 SETCOLOR A1,6,8
4034 SETCOLOR A0,A5,8
4036 SETCOLOR A2,14,8
4038 SETCOLOR 3,3,8
4040 POSITION A4,0:PRINT #6;"TODAY'S top"
4130 FOR N=A1 TO A5
4140 MIN=INT(RECORD(N)/60)
4145 SEC=RECORD(N)-MIN*60
4150 POSITION A0,2*N+1:PRINT #6;MIN;":";SEC
4160 POSITION 8,A2*N+A1:PRINT #6;NAME$((N-A1)*12+A1,N*12)
4170 NEXT N
4180 POSITION A1,23:PRINT #6;"hit START to start";
4190 RETURN
4500 REM DISPLAY TOTAL RECORDS
4510 FOR N=53248 TO 53255:POKE N,A0:NEXT N
4520 GRAPHICS A0
4530 POKE 559,62
4535 POKE 752,A1
4540 SETCOLOR A2,14,A0:SETCOLOR A1,0,8
4550 POSITION 11,A0:PRINT "TOTAL TOP RECORDS"
4560 FOR N=A1 TO 20
4570 MIN=INT(TOP(N)/60)
4580 SEC=TOP(N)-MIN*60
4585 POSITION A0,N+A1:PRINT N;"."
4590 POSITION A4,N+A1:PRINT MIN;":";SEC
4600 POSITION 13,N+A1:PRINT KEEPER$((N-A1)*12+A1,N*12)
4610 POSITION 28,N+A1:PRINT DATE$((N-A1)*A5+A1,N*A5)
4620 NEXT N
4625 POSITION 10,23:PRINT "HIT START TO START";
4630 RETURN
5000 FOR N=A1 TO A5
5030 IF REC<RECORD(N) THEN GOSUB 5100:N=11
5040 NEXT N
5050 RETURN
5100 FOR W=A5 TO N+A1 STEP -A1
5110 RECORD(W)=RECORD(W-A1)
5125 NAME$((W-A1)*12+A1,W*12)=NAME$((W-A2)*12+A1,(W-A1)*12)
5130 NEXT W
5140 RECORDMAX=RECORD(A5)
5150 RECORD(N)=REC
5155 W=N:GOSUB 5200:REM SIGN NAME
5160 NAME$((W-A1)*12+A1,W*12)=INNAME$
5170 RETURN
5200 FOR N=53248 TO 53255:POKE N,A0:NEXT N
5230 GRAPHICS 17
5231 POKE 559,62
5232 SETCOLOR A1,3,8
5233 SETCOLOR A2,A4,8
5234 SETCOLOR 3,8,8
5238 SETCOLOR A0,15,A5
```

```
5240 POSITION A2,A0:PRINT #6;"congratolation!"
5250 POSITION A0,A4:PRINT #6;"SIGN YOUR NAME PLS.";
5260 POSITION A0,20:PRINT #6;"time used      ";
5270 MIN=INT(REC/60):SEC=REC-MIN*60
5280 PRINT #6;MIN;":";SEC;
5300 POSITION 3,13
5305 INNAME$="            "
5310 COUNT=A0
5320 POKE 764,255
5400 GET #A1,Y:COUNT=COUNT+A1
5410 IF Y=155 OR COUNT=13 THEN RETURN
5412 IF Y=126 AND COUNT<>A1 THEN COUNT=COUNT-A2:POSITION COUNT+3,13:PRINT #6;" "
:POSITION COUNT+3,13:GOTO 5400
5415 IF Y=126 THEN COUNT=COUNT-A1:GOTO 5400
5420 PRINT #6;CHR$(Y);
5430 INNAME$(COUNT,COUNT)=CHR$(Y)
5440 GOTO 5400
5450 RETURN
5800 UPDATED=A1
5805 FOR N=A1 TO 20
5810 IF REC<TOP(N) THEN GOSUB 5900:N=20
5820 NEXT N
5825 GOSUB 8000
5830 RETURN
5900 FOR W=20 TO N+A1 STEP -A1
5910 TOP(W)=TOP(W-A1)
5930 KEEPER$((W-A1)*12+A1,W*12)=KEEPER$((W-A2)*12+1,(W-A1)*12)
5937 DATE$((W-A1)*A5+A1,W*A5)=DATE$((W-A2)*A5+A1,(W-A1)*A5)
5940 NEXT W
5950 TOP(N)=REC
5955 IF REC>RECORD(10) THEN W=N:GOSUB 5200:N=W
5960 KEEPER$((N-A1)*12+A1,N*12)=INNAME$
5965 DATE$((N-A1)*A5+A1,N*A5)=TODAY$
5970 RETURN
6000 FOR W=A1 TO 5
6020 POSITION A5,6:PRINT #6;"times up";
6030 FOR N=A1 TO 20
6040 SOUND A0,100,A5,15
6050 NEXT N
6060 POSITION A5,6:PRINT #6;"        ";
6070 FOR N=A1 TO 20
6080 SOUND A0,A0,A0,A0
6090 NEXT N
6100 NEXT W
6110 RETURN
7000 FOR N=A0 TO 6
7020 LOCATE N,A2,Y
7030 TIME$(N+A1,N+A1)=CHR$(Y)
7040 NEXT N
7050 FOR W=A1 TO A5
7060 POSITION A0,A2:PRINT #6;TIME$
7070 FOR N=A1 TO A5
7080 SOUND A0,60,A5,15
7090 NEXT N
7100 POSITION A0,A2:PRINT #6;"        ";
7110 FOR N=A1 TO A5
7120 SOUND A0,A0,A0,A0
7130 NEXT N
7140 NEXT W
7150 RETURN
8000 OPEN #A2,8,A0,"D1:TOPREC.DAT"
8020 FOR N=A1 TO 20
8030 PRINT #A2;TOP(N)
8033 PRINT #A2;KEEPER$((N-A1)*12+A1,N*12)
8035 PRINT #A2;DATE$((N-A1)*A5+A1,N*A5)
8040 NEXT N
8045 CLOSE #A2
8046 RETURN
9000 GRAPHICS 0:POKE 752,1
9010 SETCOLOR 2,9,0
9020 SETCOLOR 1,3,8
9030 SETCOLOR 3,3,8
9040 SETCOLOR 4,9,0
9050 PRINT "           ATARICUP  RACING"
9055 ?.:? :? :?
9060 ? "                                  "
9070 ? "                                  "
9080 ? "                                  "
9090 ? "                                  "
9100 ? "                                  "
9110 ? "                                  "
9120 ? "                                  "
9130 ? "                                  "
9140 ? "                                  "
9150 ? "                                  "
9160 ? :? :? :? :?
9170 ? "       PROGRAMMED BY BAN SZETO"
9175 ?
9180 ? "         COPYRIGHT R    1982";
9210 RETURN
```





ATARI冬季奧運會



在炎炎的夏日能到外地避暑可謂賞心樂事了，但不是人人有時間到外地。現提供三個雪地遊戲程式，都是爲ATARI而寫的。這些程式利用ATARI的某些功用，例如：SOUND，SETCOLOR等等，使遊戲聲色兼備。

程式（1）爲：下坡過旗滑雪賽。

玩者需避開三角形的障礙物，並在兩旗之間衝下，或撞倒旗也可。遊戲以旗數及時間爲記錄。

程式（2）爲：避樹大賽。

當玩者開始遊戲，圓形即下墮，玩者避開樹之後，觸及目標即繼續，而樹的數目亦增加。

程式（3）爲：雪坑滑雪下山賽。

玩者有多個難度選擇，玩者需控制在雪坑中滑下不能觸及坑壁。以時間爲準。

以上三個遊戲程式都是使用JOYSTICK 1。

用JOYSTICK控制可分爲多部分，當輸入程式後，使之運行用RUN。如開始遊戲需按FIRE，即JOYSTICK的橙色制，轉局需把JOYSTICK向上推。當遊戲完畢另開新局即按KEYBOARD的任何一制，當然是除BREAK及SYSTEM RESET。

在程式之中用了很多的圖案因未能用印字機印出，所以只好繪上，想ATARI用家可以找到的。如：程式（3）的第30句是PRINT後有一圖案，是平時的COTL 2即B一響吧了。

因各遊戲都比較簡單，所以不多加解釋，只希望提出三個好玩的程式給大家。

滑雪過旗

```
5 SETCOLOR 2,8,6:SETCOLOR 1,0,0:SETCOLOR 4,8,6
10 POKE 752,1
20 DIM A$(6):A$="||//\\"
30 PS=18:PF=0:PO=0:PB=0:C=1:FG=0:TIME=0
40 PRINT "ü"
50 POSITION 15,5:PRINT " START! "
60 POSITION 15,7:PRINT "█████"
70 POSITION 15,8:PRINT "█   █"
80 POSITION 15,9:PRINT "█   █"
90 POSITION 15,10:PRINT "◥ || ◤"
100 POSITION 2,12:PRINT "◣
110 PRINT "|--------------------------------|"
111 FOR BELL=1 TO 8:SOUND 0,10,10,12:FOR I=1 TO 15:NEXT I:SOUND 0,0,0,0:FOR I=1
TO 30:NEXT I:NEXT BELL
112 IF STRIG(0)=1 THEN 112
115 FOR LOOP=0 TO 30*15
120 S=STICK(0)
130 A=1:TIME=TIME+0.12
140 IF S>=9 AND S<=11 THEN A=3:PS=PS-1:TIME=TIME+0.05:IF PS<2 THEN PS=36
150 IF S>=5 AND S<=7 THEN A=5:PS=PS+1:TIME=TIME+0.05:IF PS>36 THEN PS=2
155 SOUND 0,40,8,4:IF A>1 THEN SOUND 0,20,8,4
160 POSITION PS,10:PRINT A$(A,A+1)
165 IF LOOP>30*15 THEN RETURN
170 IF C=14 AND PS>=PF-1 AND PS<=PF+5 THEN FG=FG+1:GOTO 200
180 IF C=5 AND PS>=PO-1 AND PS<=PO+7 THEN 1000
190 IF C=5 AND PS>=PB-1 AND PS<=PB+7 THEN 1000
200 C=C+1:IF C>15 THEN C=1
205 IF C<>1 AND C<>8 THEN 240
210 IF C=1 THEN PF=INT(RND(0)*25+5):GOTO 250
220 IF C=8 THEN PO=INT(RND(0)*29+2)
225 IF C=8 THEN PB=INT(RND(0)*29+2):IF PB<PO-9 AND PB>PO+9 THEN 225
230 GOTO 260
240 POSITION 0,23:PRINT :GOTO 310
250 POSITION 1,23:PRINT INT(LOOP/15)+1;:POSITION PF,23:PRINT "▾   ▾":GOTO 310
260 POSITION PO,23:PRINT"▲▲▲▲"
265 POSITION PB,22:PRINT"▲▲▲▲"
310 NEXT LOOP
320 FOR I=1 TO 15:GOSUB 120:POSITION 0,23:PRINT :NEXT I:GOSUB 120
330 POSITION 2,23:PRINT "◣                              ◣"
 :GOSUB 120
340 POSITION 2,23:PRINT "|--------------------------------|"
350 FOR I=1 TO 15:GOSUB 120:POSITION 0,23:PRINT :NEXT I
355 SOUND 0,0,0,0:POSITION 2,15
360 PRINT "YOU HAVE PASSED ";FG;" FRAGS."
370 PRINT "YOUR TIME IS ";TIME;" SECONDS."
380 PRINT "DO YOU WANT TO TRY AGAIN ?"
390 IF PEEK(53775)<>255 THEN RUN
400 GOTO 390
1000 FOR TONE=150 TO 225
1010 SOUND 0,TONE,12,15
1020 NEXT TONE
1030 SOUND 0,0,0,0
1040 PRINT :PRINT :PRINT "DISQUALIFIED!!!"
1050 PRINT "SO FAR YOU HAVE PASSED ";FG;" FRAGS."
1060 GOTO 380
```

避樹

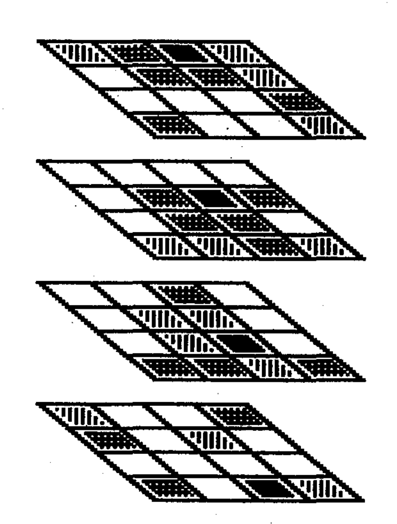
```
10 GRAPHICS 0:POKE 752,1
15 SETCOLOR 2,8,10:SETCOLOR 4,8,10:SETCOLOR 1,0,0
20 FOR I=0 TO TR
30 POSITION INT(RND(0)*39),INT(RND(0)*23):PRINT "♣"
40 NEXT I
50 H=INT(RND(0)*33)+3
60 POSITION H,21:PRINT "[ ]"
65 M=INT(RND(0)*33)+2
70 POSITION M,0:PRINT "[●]"
80 FOR I=1 TO 3
90 POSITION M,I:PRINT "   "
100 NEXT I
110 X=M+1:Y=0
115 IF STRIG(0)=1 THEN 115
120 IF STICK(0)=11 THEN 200
130 IF STICK(0)=7 THEN 270
140 SOUND 0,70,8,8
150 Y=Y+1:LOCATE X,Y,T
160 IF T=16 THEN 340
170 POSITION X,Y:PRINT "●"
180 IF Y=21 AND X>=H AND X<=H+1 THEN 460
185 IF Y=22 THEN 420
190 GOTO 120
200 SOUND 0,40,8,12
210 X=X-1:IF X<0 THEN X=38
220 LOCATE X,Y,T:IF T=16 THEN 340
230 POSITION X,Y:PRINT "● "
240 IF Y=21 AND X>=H AND X<=H+1 THEN 460
250 IF Y=22 THEN 420
260 GOTO 120
270 SOUND 0,40,8,12
280 X=X+1:IF X=39 THEN X=0
290 LOCATE X,Y,T:IF T=16 THEN 340
300 POSITION X,Y:PRINT "●"
310 IF Y=21 AND X>=H AND X<=H+1 THEN 460
320 IF Y=22 THEN 420
330 GOTO 120
```

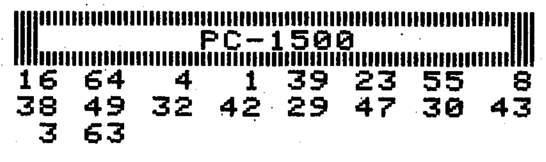
```
340 POSITION X,Y:PRINT "X"
350 FOR S=80 TO 200 STEP 2
360 SOUND 0,S,12,15:POKE 712,S
370 NEXT S:SOUND 0,0,0,0:POKE 712,138
380 POSITION 2,1:PRINT " TOO BAD!! YOU HAVE RAN INTO A TREE.":GOSUB 600
390 POSITION 2,2:PRINT " YOUR SCORE IS ";TR;"."
395 POSITION 2,3:PRINT " FROM THE TOP AGAIN."
400 IF PEEK(53775)=255 THEN 400
410 RUN
420 GOSUB 660
423 POSITION 2,18:PRINT " OOPS!!! YOU HAVE GONE TOO FAR!":GOSUB 600
425 POSITION 2,19:PRINT " YOUR SCORE IS ";TR;"."
430 POSITION 2,20:PRINT " FROM THE TOP AGAIN."
440 IF PEEK(53775)=255 THEN 440
450 RUN
460 POSITION 2,18:PRINT " VERY GOOD!!!"
461 FOR I=0 TO 5
462 SOUND 0,30,10,15:FOR II=0 TO 5:NEXT II
463 SOUND 0,30,10,0:FOR II=0 TO 5:NEXT II
464 NEXT I
465 XX=TR:GOSUB 600
470 POSITION 2,19:PRINT " YOUR SCORE NOW IS ";TR;"."
475 POSITION 2,20:PRINT " READY TO CONTINUE?"
480 IF STICK(0)=15 THEN 480
490 XX=XX+1:TR=XX
500 GOTO 10
600 TE=0
610 FOR M=TR TO 0 STEP -1
620 TE=TE+TR
630 NEXT M
640 TR=TE
650 RETURN
660 FOR I=1 TO 5:SS=I*20+20
670 FOR S=SS+80 TO SS STEP -10
680 SOUND 0,S,10,15
690 NEXT S
700 NEXT I:SOUND 0,0,0,0
710 RETURN
```

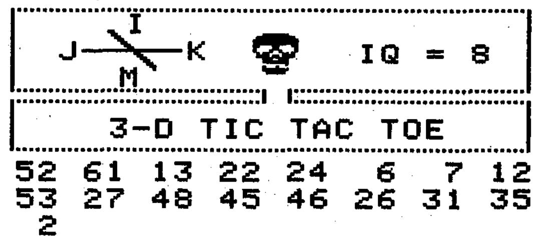
雪坑車

```
1 GRAPHICS 1+16
2 DIM RD$(13)
3 GOSUB 550
10 GRAPHICS 0:SETCOLOR 2,8,6:SETCOLOR 1,0,0:SETCOLOR 4,8,6
11 POKE 752,1:TIME=0:SS=40
12 DIM A$(6):A$="|| //\\"
13 TIME=0:POSITION 15,5:PRINT " START! "
14 POSITION 15,7:PRINT "■■■■■■■■"
15 POSITION 15,8:PRINT "■      ■"
16 POSITION 15,9:PRINT "■      ■"
17 POSITION 15,10:PRINT "◥  ||  ◤"
18 FOR U=1 TO 8:SOUND 0,20,10,8:FOR I=1 TO 20:NEXT I:SOUND 0,0,0,0:FOR I=1 TO 20
:NEXT I:NEXT U
19 IF STRIG(0)=1 THEN 19
30 PRINT "↖"
40 PR=13
50 PC=PR+5
55 COLOR 42:PLOT 0,23:DRAWTO 39,23
58 FOR LOOP=1 TO 500
60 S=STICK(0)
70 IF S>=9 AND S<=11 THEN A=3:GOTO 100
80 IF S>=5 AND S<=7 THEN A=5:GOTO 110
90 A=1:LOCATE PC,10,C1:LOCATE PC+1,10,C2:IF C1=32 AND C2=32 THEN 120
95 GOTO 500
100 TIME=TIME+0.05:LOCATE PC-1,10,C:IF C=32 THEN PC=PC-1:GOTO 120
105 GOTO 500
110 TIME=TIME+0.05:LOCATE PC+2,10,C:IF C=32 THEN PC=PC+1:GOTO 120
115 GOTO 500
120 TIME=TIME+0.1:POSITION PC,10:PRINT A$(A,A+1)
122 SOUND 0,40,8,8
123 IF A>1 THEN SOUND 0,20,8,8
125 IF LOOP>500 THEN RETURN
130 PR=PR+SGN(RND(0)-RND(0))
140 IF PR>26 THEN PR=PR-2
150 IF PR<1 THEN PR=PR+2
160 POSITION PR,23:PRINT RD$
170 NEXT LOOP
180 FOR I=1 TO 18:GOSUB 60:POSITION 0,23:PRINT :NEXT I:SOUND 0,0,0,0
185 POSITION 2,15:PRINT "CONGRATULATIONS!!!"
190 PRINT "YOUR TIME IS ";TIME;" SECONDS."
200 PRINT "DO YOU WANT TO TRY AGAIN ?                "
205 PRINT "                                          "
210 IF PEEK(53279)=6 THEN RUN
215 IF PEEK(53775)=255 THEN 210
220 PRINT "ü":GOTO 13
500 FOR SOUND=100 TO 200
510 SOUND 0,SOUND,12,15
520 NEXT SOUND:SOUND 0,0,0,0
522 POSITION 2,12
523 PRINT "                                          ":PRINT "DISQUALIFIED!!!
       "
525 PRINT "YOU HAVE FINISHED ";INT(LOOP/510*100);"% OF THE JOURNEY.":GOTO 200
550 COLOR 163:PLOT 0,0:DRAWTO 19,0:DRAWTO 19,23:DRAWTO 0,23:DRAWTO 0,0
560 POSITION 6,4:PRINT #6;"____________"
570 POSITION 6,6:PRINT #6;"downhill run"
580 POSITION 6,7:PRINT #6;"____________"
590 FOR I=0 TO 23
600 POSITION 3,I:PRINT #6;"!!"
610 SOUND 0,I*8,10,10
620 FOR J=1 TO 10:NEXT J
630 NEXT I
640 SOUND 0,0,0,0
650 POSITION 9,12:PRINT #6;"select"
660 POSITION 9,13:PRINT #6;"level"
670 POSITION 9,14:PRINT #6;"(1-5)"
680 OPEN #3,12,0,"K:"
690 GET #3,P
700 CLOSE #3
710 IF P=49 THEN RD$=".*■        ■*.":RETURN
720 IF P=50 THEN RD$=".*■       ■*.":RETURN
730 IF P=51 THEN RD$=".*■      ■*.":RETURN
740 IF P=52 THEN RD$=".*■     ■*.":RETURN
745 IF P=53 THEN RD$=".*■    ■*.":RETURN
750 GOTO 680
```









APPLE][48K PLUS

《打井遊戲》這小朋友的玩意，許多人已對此不恥一顧，然則現在打怪獸，食鬼食怪之餘，曾否想到玩井的機會？各位也可曾玩過電腦程式色的五體打井？相信任何各位一玩之後，便會對自己的IQ有所懷疑，對自已的眼力、智力，五體透視力及定力要從新估計。

因爲設計這個程式需求甚苛，餘卻強大記憶及運算度要快之外，適當的構思更爲重要，故此程式多以機械碼編寫。另外，打井的位置不能在任何限制方能發揮這遊戲的功能。（最近一外地雜誌曾登出一個打井遊戲，可惜規限多多，雖然所佔程式空間較小，但令遊戲者有捉襟見肘，不能發揮之感，有興趣的人仕請參考近期的微電腦專刊）

這裏介紹的彩色「立體打井」爲BASIC編寫（99%），花去筆者過半年的時間，並且困難重重，若非編者多度摧促下連夜趕工，可能現在仍未能與讀者見面！（編者註：此程式完全爲香港土產，值得外地雜誌參考及令港人增光）

《遊戲簡介》

這遊戲玩法非常簡單，可分爲二人對打及與電腦對奕兩種，當然對電腦對奕爲這遊戲精要之處。難度選擇有八級，可適合及訓練不同IQ之人仕。1～3專爲第一類接觸人仕而設，用以熱身。對自稱有3～D心志之仕，則可自4或5級開始。但是獸性的第6級一關能否通過則閣下自理了！至於如何「獸性」，各位一試之後自有分曉。第7及8級則較人性化。第7級爲狡猾小人，而第8級則爲犴狡而居心叵測的成年人，若跟它們週旋並稍一不慎時………總之萬事看開一點。

當IQ級數選定便可輸入姓名（眞假均可），電腦便將進取的橙色供你使用，而幽晦冷酷的綠色則屬於電腦的骷髏頭。相信各位對此頭骨有曾經相識的感受，實則抄「錄影帶謀殺案」的頭骨，故此名爲電腦謀殺案，並且雙眼閃閃發光，希望各位不致盤盤皆落索。

在二人對奕時，橙色爲先行者而綠色爲冷靜的後隨者。當一切就緒，橙色CURSOR及YOUR TURN NOW開始噩夢。（二人對奕時有綠色CURSOR），衹需用J，K作左右及I，M作上下移動至適當位置後按下RETURN便成。贏棋方式衹需如平面打井一般橫，豎或對角（當然是立體表現方式）連中四元便可。雖然各位未必爲英籍人仕，當你勝出時會代你奏出女皇國歌，讓你光宗耀祖一番。若當骷髏頭勝出時可自跳躍歡欣的音符中感受到他的快樂，但筆者不付任何損壞的責任。當在遊戲之際聽到「天鵝湖」中戰役一役樂章時笑也無謂，因爲骷髏已找到「西頭蛇」，故此輸實！若中途停機當然可按下ESC鍵，然則實太丟臉，至於事後效果，請多多包涵！

《程式原理》

在屏幕上的四個4×4棋盤實爲一個64正立方體連成的大正理方體，故此各位需將每位盤視作一個正方立體，自此察覺到棋盤上有64點，18塊平面，76條綫。整體有八隻角，中心8點爲心，6個表面中心4點（共24點）爲面，其他（共24點）爲邊。每個角或心點上約有7綫穿過（有7關方向），而每個面或邊點均有4綫穿過，可見角心攻守力較強，各位藉此易掌握得多！

當電腦思考時，它會以角、心、邊、面上已佔領的情況下逐計眞評分，找出最高分的一着將挑戰者攻擊。

《程式細節》

程式中行數少於1φφφ及大於1φφφ的均是副程式其中數個非常實用，不妨抄錄至其他程式中使用，略述如下：

SCREEN DUMP副程式，在3φφφ至37φ，在此以EPSON印字機作標準。（因爲在實用不同的印字機時，令有不同的需要，故此未能博愛，請各位忍受一下）33φ行中有將所有綠色以另一形狀重繪一次的現象，衹因橙綠色在印字機而言並無分別，此法使混淆處分開，印發後綠色成爲一點點，橙色成爲間條，而白色則將整格塗黑。SCREEN DUMP後便可永世保留光榮紀錄。當印字機工作時，按下CTRL－C可將印字工作

停止且不會擾亂程式。

但在其他情況下按CTRL－C會HANG機直至RESET或OFF。筆者有一難體，無法第一次SCREEN DUMP時印出畫面，代之是一堆垃圾，但以後便乖乖地印出畫面，實在丈八金剛，希望遇到高明賜教，感甚！謝甚！

LORES SHAPE DRAW副程式自6φφ至69φ，能將HIRES SHAPE以LORES繪出換言之可用SHAPE TABLE創作LORES SHAPE，不再是HIRES的專刊，但要注意到這副程式始自64φ，而63φ爲二階副程式，需要3個參數，即LORES的，起始LOCATION（Q=32768），X及Y軸座標（在此爲XX%及YY%），在監視器上（或TV）所見的LORES GRAPHIC藉此產生，故此速度稍減，但不用逐點描出，當然亦不會計到手軟及佔用大量變數記憶空間。雖然此程式好處甚多，但本身沒有任何保護設施，當X%或Y%少於零或大於39時有ERROR，故此抄這程式切記加入保護。（本程式中沒有此等情況，故可省則省。）

READ DATA LINE副程式 祇38φ一行雖嫌簡單，實爲精彩之處，且耐心看下去。本程式雖有近二千DATA，卻無任何RESTORE，那麼怎樣指定數據呢？這個困難險些不能解決，在花去數十工作小時後才能圓滿完工，方法是將APPLESOFT的假密碼（PSEUDO－CODE）拆解，方可隨心所以地讀數據，至於假密碼的形式（FORMAT）頗爲煩複，不擬討論，技術方面解釋如下：

在APPLE PAGE ZERO中有兩個位置儲最新DATA的地址，分別爲125 及 126，儲起地址碼的低位及高位字元，若POKE入適當的數值便可讀入指定數據，免卻FOR－LOOP所浪費的時間及程式空間。經實驗後是將125及126POKE入DATA這關鍵字所在的地址。DATA的TOKEN爲&H83，可藉此解決讀數據，但需時間找數據地址，若你將DATA置在程式末時，那麼，皇天庇佑！

樂曲及機械碼副程式。 這程式稱爲99%APPLE－SOFT BASIC，是因爲這裏不足1%的6502 M．C．程序而名。在程式開始時便被POKE入&H3φ2至&H33A。 &H3φ2至&H327演奏音樂，輔以程式335φ至341φ數據奏出。&H328至&H33A爲音響效果。每個程序均需兩個參數，一爲頻率，二爲音長。&H3φ2程式中頻率參數在&H3φ1，音長在&H3φφ，而&H328程式則以&H6及&H7，可是第二部份程式中頻率及音長效果頗爲特別，須自己捉摸，不能筆墨形容。對於祇知POKE／PEEK－16336的朋友們想奏音樂者，此程式能令君夙願得償。若嫌蘋果沒有中氣，可以在卡式出口接出聲音加以放大，祇需將3φφφ行DATA內兩個48改成32便成，但萬勿改錯，否則九成HANG機。以下爲各行DATA的歌曲名稱以資參考：

（每行第一個數爲音符總數，以後是頻率及音長數據的梅花間竹排列。）

335φ：GOD SAVE THE QUEEN
336φ：UNKNOWN
337φ：ODE OF JOY
338φ：SWAN LIKE
339φ及34φφ：UN BEMC，UN ARBRE，UNE RUE
341φ：DOWN THE ROAD

至於其他的DATA句闡釋如下：

3φ1φ：八隻角的位置
3φ2φ～3φ4φ：18平面資料
31φφ～319φ：各IQ度（LEVEL）所需資料
32φφ：核對用資料
321φ～334φ：SHAPE TABLE

```
]CALL-151

*302LL

0302-   AC 01 03    LDY   $0301
0305-   AE 01 03    LDX   $0301
0308-   A9 04       LDA   #$04
030A-   20 A8 FC    JSR   $FCA8
030D-   AD 30 C0    LDA   $C030
0310-   E8          INX
0311-   D0 FD       BNE   $0310
0313-   88          DEY
0314-   D0 EF       BNE   $0305
0316-   CE 00 03    DEC   $0300
0319-   D0 E7       BNE   $0302
031B-   60          RTS
031C-   20 20 70    JSR   $7020
031F-   08          PHP
0320-   18          CLC
0321-   D8          CLD
0322-   88          DEY
0323-   08          PHP
0324-   A0 A0       LDY   #$A0
0326-   10 38       BPL   $0360
0328-   AD 30 C0    LDA   $C030
032B-   88          DEY
032C-   D0 04       BNE   $0332
032E-   C6 07       DEC   $07
0330-   F0 08       BEQ   $033A
0332-   CA          DEX
0333-   D0 F6       BNE   $032B
0335-   A6 06       LDX   $06
0337-   4C 28 03    JMP   $0328
033A-   60          RTS
```

《程序輸入》

由於RESTORE DATA時用了茅招以加快程序運行速度，所以程式長度絕不可改變，只要短了一BYTE便會HANG機，因此打入時必要（絕對要！！）一千萬小心，不能有錯，另須注意之處闡述如下：

（1）所有REM必要照原程式打入，長度及空位亦是，但REM後的第一個空位是APPLE在LIST

時自動加上故毋須打入，故ＲＥＭ及隨後文字連起打入。如４１φ ＲＥＭ　ＷＩＮＤＯＷ　ＣＬＥＡＲＩＮＧ，只在Ｗ及Ｃ間有空位而Ｍ及Ｗ間沒有。

（２）所有字串（ＬＩＴＥＲＡＬ　ＳＴＲＩＮＧ）內要補足空位，並要十分小心留意字串間空位數（因有時不祇一個空位）。

（３）在程式中有時出現ＰＯＫＥ１２５，φ４８（如１７２φ 行），該φ 十分重要，不能忽略。若果不信，一試便知。

（４）在１１φφ 及１２φφ 行中的一對倒轉中括號（］〔）一個是ＣＨＲ＄（９１），另一個是ＣＨＲ＄（９３），在標準ＡＰＰＬＥ鍵盤上只可利用ＳＨＩＦＴ－Ｍ產生ＣＨＲ＄（９３），而ＣＨＲ＄（９１）要印出後用編輯（ＥＤＩＴ）功能去ＣＯＰＹ，對於不熟譜ＥＤＩＴ的朋友可用兩個Ｉ代替，但沒有那麼美觀。

（５）因爲很多句的長度僅能輸入ＢＵＦＦＥＲ中，故此有加ＳＰＡＣＧ習慣的人肯定會ＢＵＦＦＥＲ　ＦＵＬＬ（同時ＰＲＩＮＴ必要用？代替）。

祇要導以上五點應沒有問題，在輸入後可打入ＰＲＩＮＴ　ＰＥＥＫ（１７５）＋ＰＥＥＫ（１７６）＊２５６，如果答案是１５７３９的話，恭喜！！你已第一關及格，再打入ＰＲＩＮＴ　ＰＥＥＫ（１１２φφ），若答案是１３１的話，表示你的輸入大致上沒有問題，若有任何錯誤的話———很不幸！重新並耐心地翻ＣＨＥＣＫ罷，除此之外，別無他法！改正後再檢驗直至正確。

切記！！不可即時運行程式，必須先ＳＡＶＥ入磁碟或磁帶中方可ＲＵＮ，否則程式不論成敗均會踏浪而去，與黃鶴說別（即是ＮＥＷ咗！）這程式長度接近１４Ｋ，沒有錯誤甚難，但只要有耐性，應可順利完成。

總結：

筆者在寫此程式中發覺到ＡＰＰＬＥＳＯＦＴ特色縱使不算空前，亦算絕後，正如ＰＲＩＮ１５＊φ．２是３及ＰＲＩＮ　ＩＮＴ（１５＊φ．２）居然是２，此等進位誤差會浪費了筆者７φ 個工作小時去除臭蟲哩！可眞寃枉。

此遊戲中電腦並非不敗者，若你能以冷靜而靜醒的三維頭腦便能過關斬將，所向披靡。但筆者原先可將它幻化成所向無敵的超人，可是在圖形，時間以至樂趣三大方面均有極大折拉呢！（筆者曾用ＡＴＡＲＩ的硬件ＣＡＲＴＲＩＤＧＥ機械語曾需時五分鐘）而且要避開ＨＩＲＥＳ，必要使用ＳＰＬＩＴＴＩＮＧ技巧，可謂相當驚人！

筆者使用ＡＰＰＬＥ九月之中，花六月寫這程式及其姊妹作（ＰＣ－１５００版本）所領悟如下：

（１）寫程式最好是兩人，互相激勵。

（２）ＢＡＳＩＣ易學易精，雖不如ＰＡＳＣＡＬ等語言，但ＢＡＳＩＣ所需數學根底毋須深厚，適合初次與電腦接觸及溝通之用。

（３）ＡＰＰＬＥ沒有磁碟機，尤如廢去右手（如ＰＣ１５００沒有ＲＡＭ及ＣＥ－１５０般）

（４）寫程式前要做足準備工夫（如畫好結構圖等），否則程式易變鷄肋。

（５）主動多參閱書藉，常分析他人作品，自然大有進步。

（６）想要進步快，要一日廿四小時想著程式，祇在吃飯時例外。

（７）最後，在設計ＰＣ１５００程式後感到ＰＣ１５００是部神奇機器，尤其是它的ＣＥ－１５０，除卻價廉物美外，簡直比想像中還好，更滿意。

編者言：

本程式除趣味性圖案及控制自如外，各種聲嚮爲本人讚賞，另外在十數秒內計眞達一千八百多次的速度（並包括發聲、貶眼及衆多ＦＯＲ－ＬＯＯＰ的功能）與及純本港人的構思，實屬港人極出色程式之中一個代表。

此程式的姊妹作爲ＰＣ－１５００版本，長度不足４Ｋ，適用於ＣＥ１５１（４Ｋ　ＲＡＭ）的人仕。本人現已情商作者刊於今月微電腦專刊中，作爲對讀者的一份禮物。

```
ARRAY CONFIGURATION TABLE OF APPLESOFT VERSION
==============================================

NAME            PURPOSE
----            -------

DX%(64)         X-AXIS POSITION OF THE FRAME
DY%(64)         Y-AXIS POSITION OF THE FRAME
M%(64)          WHETHER-THE-POSITION-IS OCCUPIED INDICATOR
BA(64)          MARK ACCUMULATOR
B%(32)          STORE THE HIGHEST-MARKED POSITIONS
L%(5)           HOLD THE CONDITIONS DATA (LINE 3200)
Z(8)            DATA OF THE SELECTED LEVEL
RD%(8)          TEMPORARYLY STORE THE CURRENT CALCULATING LINE'S POSITION
```

```
]POKE1657,136

]LIST

10  REM THREE-DIMENSIONAL TIC-TAC-TOE VERSION 78.3
20  REM BY CHARLES C.H.HUNG & PLUTO M.K.NG, A.D. 1982
30  REM MIND-TORTURER
40  REM 99% APPLESOFT, 48K SYSTEM ONLY
50  REM AP-II, ORANGE II, PEAR II, SP-201 COMPATIBLE
60  REM
70  REM INITIALIZE SYSTEM
80  REM
90  LOMEM: 24576: HIMEM: 32768: CLEAR : POKE 216,254: POKE 1012,0: POKE 9651,191: FOR I = 770 TO 826: READ X%: POKE I,X%: NEXT : GOTO
    1000
200  REM
210  REM STEP NO.
220  REM
230 X% = (SP% -  INT (SP% / 8) * 8) * 21 + 119:Y% =  INT (SP% / 8) * 10 + 30 + (M%( VAL (P$(3))) = 1) * 102:P$(3) = " " + P$(3):D
    U$ =  RIGHT$ (P$(3),2): HCOLOR= 3: GOSUB 540: RETURN
300  REM
310  REM SCREEN DUMP (ESPON ONLY)
320  REM
330  HCOLOR= 3: FOR A = 1 TO P%: IF M%(A) = 1 THEN  DRAW 3 AT DX%(A),DY%(A)
340  NEXT :DU$ = SB$: GOSUB 530: FOR A = 20 TO 1 STEP  - 1: POKE 6,50: POKE 7,A: CALL W%: NEXT
350 P4% = 2: POKE R%,0: WAIT S%,128:X% =  PEEK (S%): IF X% = 197 THEN P4% = P4% + P%:X% = 206
360  IF X% <  > 206 THEN  POKE 6,255: POKE 7,255: CALL W%: GOTO 350
370 DU$ = SC$: GOSUB 530: PR# 1: POKE 1913,P4%: PRINT  CHR$ (17);: PR# 0: PRINT : RETURN
400  REM
410  REM WINDOW CLEARING
420  REM
430  HCOLOR= 0: FOR A = 1 TO 5: HPLOT 126 + A,Y% - 9 + A TO 272 - A,Y% - 9 + A: HPLOT  TO 272 - A,Y% + 2 - A: HPLOT  TO 126 + A,Y
    % + 2 - A: HPLOT  TO 126 + A,Y% - 9 + A: NEXT : RETURN
500  REM
510  REM HIRES SHAPE DRAW
520  REM
530 Y% = 116: GOSUB 400:X% = 200 -  INT ( LEN (DU$) * 7 / 2): HCOLOR= 3
540  POKE 232,0: POKE 233,128: FOR A = 1 TO  LEN (DU$):BU% =  ASC ( MID$ (DU$,A,1)) - 31: DRAW BU% AT X%,Y%:X% = X% + 7: NEXT : POKE
    232,16: POKE 233,131: RETURN
600  REM
610  REM LORES SHAPE DRAW
620  REM
630 ED% = P4% / 2:X% = X% + ( - 1) ^ ED% * (P4% - ED% * 2):Y% = Y% - ( - 1) ^ ED% * (P4% + 1 -  INT ((P4% + 1) / 2) * 2): RETURN

640  FOR A = 1 TO  LEN (DU$):Y% = YY%:X% = XX% + A * 6 - 6:BU% =  ASC ( MID$ (DU$,A,1)) - 31:ST =  PEEK (Q + BU% * 2) +  PEEK (Q +
    BU% * 2 + 1) * 256 + Q:ED = ST + 1000: FOR B = ST TO ED:PL% =  PEEK (B): IF PL% = 0 THEN B = ED: GOTO 690
650 P1% = PL% / P%:P2% = (PL% - P1% * P%) / 8:P3% = PL% - P1% * P% - P2% * 8: IF P3% > 3 THEN  PLOT X%,Y%
660 P4% = P3%: GOSUB 630: IF  NOT (P1% + P2%) THEN 690
670  IF P2% > 3 THEN  PLOT X%,Y%
680 P4% = P2%: GOSUB 630: IF P1% > 0 THEN P4% = P1%: GOSUB 630
690  NEXT : NEXT : RETURN
700  REM
710  REM REFRESH THE DRAWING
720  REM
730  FOR A = 1 TO P%: IF M%(A) THEN  HCOLOR= M%(A): DRAW 2 AT DX%(A),DY%(A)
740  NEXT : RETURN
800  REM
810  REM READ SPECIFIED DATA LINE
820  REM
830 BU% = 11200 + (LE% - 1) * 31: POKE 126,BU% / 256: POKE 125,BU% -  PEEK (126) * 256: FOR A = 0 TO 8: READ Z(A): NEXT : RETURN

1000  REM
1010  REM PREPARATIONS
1020  REM
1030  TEXT : HOME
1050  FOR I = 1 TO 29:P$(1) = P$(1) + " ": NEXT :P$(0) = "THREE-DIMENSIONAL TIC-TAC-TOE": VTAB 20: HTAB 14: PRINT "COPYRIGHT  198
    2": PRINT : HTAB 6: PRINT "CHARLES C.H.HUNG & PLUTO M.K.NG": COLOR= 13: HLIN 5,35 AT 12: HLIN 5,35 AT 16
1060  FOR I = 1 TO 3: FOR J = 1 TO 29:P$(1) =  RIGHT$ (P$(1),28) +  MID$ (P$(0),J,1): VTAB 8: HTAB 7: PRINT P$(1);: FOR K = 1 TO
    50: NEXT : POKE 6,11: POKE 7,5: CALL 816: NEXT :P$(1) =  RIGHT$ (P$(1),28) + " ": NEXT
1070  POKE 125,009: POKE 126,45: FOR I = 32768 TO 33825: READ BU%: POKE I,BU%: NEXT
1080  CLEAR :P% = 64:Q = 32768:R% =  - 16368:S% =  - 16384:T% = 770:U% = 769:V% = 768:W% = 808
1090 B$ =  CHR$ (7):S0$ = "COWARD":S1$ = "YOUR TURN NOW":S2$ = "I'M THINKING":S3$ = "WHO FIRST ? (P/C)":S4$ = "PHOOEY! YOU WIN":S
    5$ = "HEE HEE,YOU LOSE":S6$ = "ANOTHER GAME ?":S7$ = "IQ = ?"
1100 S8$ = "(1-8)":S9$ = "THE GAME TIED":SA$ = "SCREEN DUMP ?":SB$ = "(E)NLARGE,(N)ORMAL":SC$ = "3-D TIC-TAC-TOE":SD$ = "APPLE ][
    48K PLUS"
1110  DIM DX%(P%),DY%(P%),M%(P%),BA(P%),B%(32),Z(8),RD%(8),L%(5),P$(3)
1120  REM
1130  REM START
1140  REM
1150  TEXT : HOME : HGR2 : SCALE= 1: ROT= 0: HCOLOR= 5: FOR I = 2 TO 21: HPLOT 119,I TO 279,I: NEXT : HCOLOR= 1: FOR I = 169 TO 1
    88: HPLOT 119,I TO 279,I: NEXT : HCOLOR= 3
1160  HCOLOR= 2: HPLOT 196,99 TO 118,99: HPLOT  TO 118,69: HPLOT  TO 278,69: HPLOT  TO 278,99: HPLOT  TO 204,99: HPLOT  TO 204,10
    3: HPLOT  TO 278,103: HPLOT  TO 278,121: HPLOT  TO 118,121: HPLOT  TO 118,103: HPLOT  TO 196,103: HPLOT  TO 196,99
1170  HCOLOR= 3: HPLOT 140,84 TO 170,84: HPLOT 148,77 TO 162,91: POKE 232,0: POKE 233,128: DRAW 43 AT 132,87: DRAW 42 AT 153,77: DRAW
    44 AT 172,87: DRAW 46 AT 151,97: POKE 232,16: POKE 233,131: DRAW 4 AT 199,84
1180 X% = 15:Y% = 15:BU% = 0: FOR I = 0 TO 3: FOR J = 0 TO 3: FOR K = 0 TO 3:BU% = BU% + 1:DX%(BU%) = X% + K * 16:DY%(BU%) = Y%: DRAW
    1 AT DX%(BU%),DY%(BU%): NEXT :X% = X% + 9:Y% = Y% + 9: NEXT :X% = 15:Y% = Y% + 9: NEXT
1190 DU$ = SC$: GOSUB 530: POKE 125,154: POKE 126,59: READ ED%: FOR I = 1 TO ED%: READ P1%,P2%: POKE U%,P1%: POKE V%,P2%: CALL T%
    : NEXT :DU$ = "1 OR 2 PLAYER": GOSUB 530
1200  POKE R%,0: WAIT S%,128:X% =  PEEK (S%): IF X% = 178 THEN  HCOLOR= 3:DU$ = "APPLE":X% = 228:Y% = 79: GOSUB 540:DU$ = "][":X%
    = 239:Y% = 87: GOSUB 540:DU$ = "48K PLUS":X% = 218:Y% = 95: GOSUB 540: GOTO 1870
1210  IF X% <  > 177 THEN  POKE 6,255: POKE 7,255: CALL W%: GOTO 1200
1220 Y% = 182: GOSUB 400:DU$ = SD$:X% = 141: HCOLOR= 3: GOSUB 540:Y% = 15: GOSUB 400:DU$ = "WHAT'S YOUR NAME": GOSUB 530: POKE 23
    2,0: POKE 233,128: FOR I = 1 TO 15: POKE 6,150: POKE 7,20: CALL W%
1230  POKE R%,0: WAIT S%,128:X% =  PEEK (S%) - 128: IF X% = 13 AND  LEN (P$(0)) THEN I = 15: NEXT : GOTO 1260
1240  IF X% < 32 THEN  POKE 6,255: POKE 7,255: CALL W%: GOTO 1230
1250  DRAW X% - 31 AT I * 7 + 141,15:P$(0) = P$(0) +  CHR$ (X%): NEXT
1260  TEXT : HOME : GR : COLOR= 12:DU$ = S7$:XX% = 1:YY% = 15: GOSUB 640:DU$ = S8$:XX% = 4:YY% = 30: GOSUB 640: FOR I = 1 TO 20: POKE
    6,100: POKE 7,I: CALL W%: NEXT
1270  POKE R%,0: FOR I =  - 255 TO  - 1:X% =  PEEK (S%): IF X% < 177 OR X% > 184 THEN  NEXT : GOTO 1270
1280 LE% = X% - 176:X% =  RND (I): GOSUB 830: POKE 49239,0: POKE 49234,0: POKE 49237,0: HCOLOR= 3:DU$ = "IQ = " +  STR$ (LE%):X% =
    225:Y% = 88: GOSUB 540: POKE 125,246: POKE 126,44: FOR I = 0 TO 5: READ L%(I): NEXT :FF% = 1:DU$ = S3$: GOSUB 530
1290  FOR I = 1 TO 20: POKE 6,250: POKE 7,I: CALL W%: NEXT
1300  POKE R%,0: WAIT S%,128:X% =  PEEK (S%): IF X% = 208 THEN FF% = 0:Y% = 116: GOSUB 400: GOTO 1350
1310  IF X% <  > 195 THEN  POKE 6,255: POKE 7,255: CALL W%: GOTO 1300
1320 Y% = 116: GOSUB 400
1330  POKE 125,122: POKE 126,38:X% =  INT ( RND (1) * 8 + 1): FOR I = 1 TO X%: READ P3%: NEXT : IF M%(P3%) THEN 1330
1340  GOTO 1780
1350  REM
1360  REM PLAYER'S ORDER
1370  REM
1380  HCOLOR= 3:X% = 153:DU$ = S1$:Y% = 116: GOSUB 540:X% =  FRE (0): FOR I = 1 TO P%: IF M%(I) THEN  NEXT
1390  HCOLOR= 5: DRAW 1 AT DX%(I),DY%(I)
1400  POKE R%,0: WAIT S%,128:X% =  PEEK (S%) - 128: HCOLOR= 3: DRAW 1 AT DX%(I),DY%(I)
1410  IF X% = 13 THEN M%(I) = 5: HCOLOR= M%(I): DRAW 2 AT DX%(I),DY%(I): GOSUB 700:P$(3) =  STR$ (I): GOSUB 200: GOTO 1460
1420  IF X% = 27 THEN  TEXT : HOME : GR :DU$ = S0$: COLOR= 9:XX% = 1:YY% = 22: GOSUB 640: POKE 125,235: POKE 126,60: READ ED%: FOR
    J = 1 TO ED%: READ P1%,P2%: POKE U%,P1%: POKE V%,P2%: CALL T%: NEXT : POKE 49239,0: POKE 49234,0: POKE 49237,0: GOTO 2100
1430 BU% = (X% = 75) - (X% = 74) + (X% = 77) * 4 - (X% = 73) * 4 + (X% = 21) * 16 - (X% = 8) * 16: IF  NOT BU% THEN  POKE 6,255: POKE
    7,255: CALL W%: GOTO 1390
1440 I = I + BU%:I = I + (I < 1) * P% - (I > P%) * P%: IF  NOT M%(I) THEN 1390
1450  GOTO 1430
1460  REM
1470  REM START THINKING
1480  REM
1490  HCOLOR= 0:DU$ = S1$:X% = 153:Y% = 116: GOSUB 540: HCOLOR= 3:X% = 157:DU$ = S2$: GOSUB 540: POKE 125,122: POKE 126,38:RL = 0
    : FOR I = 0 TO P%:BA(I) = 0: NEXT : IF SP% > 20 OR Z(6) THEN  POKE 125,065: POKE 126,40: GOTO 1540
1500  FOR I = 1 TO 8: READ RD%(I): IF  NOT M%(RD%(I)) THEN BA(RD%(I)) = BA(RD%(I)) + 5
1510  NEXT : FOR I = 1 TO 18: FOR J = 1 TO 8: READ RD%(J): NEXT
1520 P1% = M%(RD%(1)) + M%(RD%(2)) + M%(RD%(3)) + M%(RD%(4)):P2% = M%(RD%(5)) + M%(RD%(6)) + M%(RD%(7)) + M%(RD%(8)): IF (P1% < 1
    5 OR P2% -  INT (P2% / 5) * 5 >  INT (P2% / 5)) AND P2% <  > 15 THEN  NEXT : GOTO 1540
1530  FOR J = 5 TO 8:BA(RD%(J)) = BA(RD%(J)) + (P1% + P2% + (P2% = 15) * 40) * (M%(RD%(J)) = 0): NEXT : NEXT
1540 OE% = 0: FOR I = 1 TO 76:OE% =  NOT OE%: HCOLOR= OE% * 3: DRAW 5 AT 199,84:BU% = 0: FOR J = 1 TO 4: READ RD%(J):BU% = BU% +
    M%(RD%(J)): NEXT : IF BU% > 4 AND BU% <  >  INT (BU% / 5) * 5 THEN 1600
1550  IF BU% = 20 THEN DU$ = S4$: GOSUB 530:BU% = 5: GOTO 2050
1560  IF BU% = 3 THEN RL = 0: FOR J = 1 TO 4:RL = RL * 100 + RD%(J): NEXT : GOTO 1600
1570  FOR J = 0 TO 5: IF BU% <  > L%(J) THEN  NEXT
1580  FOR K = 1 TO 4: IF  NOT M%(RD%(K)) THEN BA(RD%(K)) = BA(RD%(K)) + Z(J)
1590  NEXT :J = 6: NEXT
1600  POKE U%,179 + I: POKE V%,OE% + 1: CALL T%: NEXT
1610  REM
```

```
1620  REM CHECK POINTS' MARK
1630  REM
1640  IF  NOT RL THEN 1680
1650 DU$ = S5$: GOSUB 530: HCOLOR= 2:BU$ = "0" +  STR$ (RL):BU$ =  RIGHT$ (BU$,8): FOR I = 1 TO 4:RD%(I) =  VAL ( MID$ (BU$,2 * I
   - 1,2))
1660  IF  NOT M%(RD%(I)) THEN M%(RD%(I)) = 1: HCOLOR= 2: DRAW 2 AT DX%(RD%(I)),DY%(RD%(I)):P$(3) =  STR$ (RD%(I)): GOSUB 200:BU% =
   1
1670  NEXT : GOTO 2050
1680 HI = .1:BU% = 0:Y% = 0: FOR I = 1 TO P%: IF BA(I) -  INT (BA(I)) > .15 THEN Y% = I
1690  IF BA(I) > HI THEN HI = BA(I):BU% = 0
1700  IF BA(I) = HI THEN B%(BU%) = I:BU% = BU% + 1
1710  NEXT : IF HI < 1 THEN 1860
1720  IF Y% AND HI < Z(5) THEN P3% = Y%: POKE 125,048: POKE 126,59: READ ED%: FOR I = 1 TO ED%: READ P1%,P2%: POKE U%,P1%: POKE V
   %,P2%: CALL T%: NEXT : GOTO 1780
1730  IF SP% < (3 + FF%) AND HI < 50 THEN 1330
1740  REM
1750  REM COMPUTER'S ORDER
1760  REM
1770 P3% = B%( INT ( RND (1) * BU%))
1780 M%(P3%) = 1: FOR I = 1 TO P3%: IF  NOT M%(I) THEN  HCOLOR= 1: DRAW 1 AT DX%(I),DY%(I): POKE U%,256 - I: POKE V%,2: CALL T%: HCOLOR=

   3: DRAW 1 AT DX%(I),DY%(I)
1790  NEXT : FOR I = 1 TO 3: HCOLOR= 2: DRAW 2 AT DX%(P3%),DY%(P3%): PRINT B$: HCOLOR= 1: DRAW 2 AT DX%(P3%),DY%(P3%): FOR J = 1 TO
   2 * P%: NEXT : NEXT
1800  HCOLOR= 0:X% = 157:Y% = 116:DU$ = S2$: GOSUB 540:P$(3) =  STR$ (P3%): GOSUB 200: GOSUB 700
1810  REM
1820  REM TIED ?
1830  REM
1840 SP% = SP% + 1: IF SP% = Z(7) THEN LE% = Z(8): GOSUB 830
1850  IF SP% < 32 THEN 1380
1860 DU$ = S9$: GOSUB 530: POKE 125,121: POKE 126,58: READ ED%: FOR I = 1 TO ED%: READ P1%,P2%: POKE U%,P1%: POKE V%,P2%: CALL T%
   : NEXT : GOTO 2100
1870  REM
1880  REM 2 PLAYER ROUTINE
1890  REM
1900  FOR I = 1 TO 2:DU$ = "NAME OF PLAYER " +  STR$ (I): GOSUB 530: POKE 232,0: POKE 233,128:Y% = 15 + (I - 1) * 167: GOSUB 400:
    FOR J = 1 TO 15: HCOLOR= 3: POKE 6,150: POKE 7,20: CALL W%
1910  POKE R%,0: WAIT S%,128:X% =  PEEK (S%) - 128: IF X% = 13 AND  LEN (P$(I)) THEN J = 15: GOTO 1940
1920  IF X% < 32 THEN 1910
1930  DRAW X% - 31 AT J * 7 + 141,I * 167 - 152:P$(I) = P$(I) +  CHR$ (X%)
1940  NEXT : NEXT : POKE 232,16: POKE 233,131: FOR I = 1 TO 32: FOR J = 5 TO 1 STEP  - 4:DU$ = P$(2 - (J = 5)) + " TURN": GOSUB 5
   30:X% =  FRE (0): FOR K = 1 TO P%: IF M%(K) THEN  NEXT : GOTO 1860
1950 P3% = K:K = P%: NEXT
1960  HCOLOR= J: DRAW 1 AT DX%(P3%),DY%(P3%)
1970  POKE R%,0: WAIT S%,128:X% =  PEEK (S%) - 128: HCOLOR= 3: DRAW 1 AT DX%(P3%),DY%(P3%)
1980  IF X% = 13 THEN M%(P3%) = J: HCOLOR= J: DRAW 2 AT DX%(P3%),DY%(P3%): GOSUB 700:P$(3) =  STR$ (P3%): GOSUB 200:DU$ = SC$: GOSUB
   530: GOTO 2030
1990  IF X% = 27 THEN 1860
2000 BU% = (X% = 75) - (X% = 74) + (X% = 77) * 4 - (X% = 73) * 4 + (X% = 21) * 16 - (X% = 8) * 16: IF  NOT BU% THEN  POKE 6,255: POKE
   7,255: CALL W%: GOTO 1960
2010 P3% = P3% + BU%:P3% = P3% + (P3% < 1) * P% - (P3% > P%) * P%: IF  NOT M%(P3%) THEN 1960
2020  GOTO 2000
2030  POKE 125,065: POKE 126,40: FOR K = 1 TO 76:BU% = 0: FOR L = 1 TO 4: READ RD%(L):BU% = BU% + M%(RD%(L)): NEXT : IF BU% = 20 OR
   BU% = 4 THEN DU$ = P$(2 - (J = 5)) + " WIN": GOSUB 530:BU% = J: GOTO 2050
2040  NEXT : NEXT :SP% = SP% + 1: NEXT : GOTO 1860
2050  REM
2060  REM END
2070  REM
2080  POKE 125,(BU% = 5) * 48 + (BU% = 1) * 147: POKE 126,57: READ ED%: FOR I = 1 TO ED%: READ P1%,P2%: POKE U%,P1%: POKE V%,P2%:
    CALL T%: NEXT
2090  FOR I = 1 TO 3: FOR J = 1 TO 4: HCOLOR= 2: DRAW 2 AT DX%(RD%(J)),DY%(RD%(J)): PRINT B$: HCOLOR= BU%: DRAW 2 AT DX%(RD%(J)),
   DY%(RD%(J)): NEXT : FOR J = 1 TO 100: NEXT : NEXT : HCOLOR= 3: FOR I = 1 TO 4: DRAW 2 AT DX%(RD%(I)),DY%(RD%(I)): PRINT B$: NEXT

2100 DU$ = SA$: GOSUB 530: FOR I = 1 TO 20: POKE 6,25: POKE 7,I: CALL W%: NEXT
2110  POKE R%,0: WAIT S%,128:X% =  PEEK (S%): IF X% = 217 THEN  GOSUB 300:X% = 206
2120  IF X% <  > 206 THEN  POKE 6,255: POKE 7,255: GOTO 2110
2130 DU$ = S6$: GOSUB 530: FOR I = 20 TO 1 STEP  - 1: POKE 6,25: POKE 7,I: CALL W%: NEXT
2140  POKE R%,0: WAIT S%,128:X% =  PEEK (S%): IF X% = 217 THEN  TEXT : HOME : GOTO 1080
2150  IF X% <  > 206 THEN  POKE 6,255: POKE 7,255: CALL W%: GOTO 2140
2160  TEXT : HOME : POKE R%,0: END
3000  DATA 172,1,3,174,1,3,169,4,32,168,252,173,48,192,232,208,253,136,208,239,206,0,3,208,231,96,32,32,112,8,24,216,136,8,160,16
   0,16,56,173,48,192,136,208,4,198,7,240,8,202,208,246,166,6,76,40,3,96
3010  DATA 1,4,13,16,49,52,61,64
3020  DATA 1,4,13,16,6,7,10,11,1,4,49,52,18,19,34,35,1,4,61,64,22,23,42,43,1,13,49,61,21,25,37,41,1,13,52,64,22,26,39,43,1,16,49.
   64,22,27,38,43
3030  DATA 4,13,52,61,23,26,39,42,4,16,49,61,23,27,38,42,4,16,52,64,24,28,40,44,13,16,49,52,26,27,38,39,13,16,61,64,30,31,46,47,4
   9,52,61,64,54,55,58,59
3040  DATA 17,20,29,32,22,23,26,27,33,36,45,48,38,39,42,43,2,14,50,62,22,26,38,42,3,15,51,63,23,27,39,43,5,8,53,56,22,23,38,39,9,
   12,57,60,26,27,42,43
3050  DATA 1,22,43,64,1,21,41,61,2,22,42,62,3,23,43,63,4,24,44,64,4,23,42,61,1,18,35,52,5,22,39,56,9,26,43,60,13,30,47,64,1,17,33
   ,49,2,18,34,50,3,19,35,51,4,20,36,52,5,21
3060  DATA 37,53,6,22,38,54,7,23,39,55,8,24,40,56,9,25,41,57,10,26,42,58,11,27,43,59,12,28,44,60,13,29,45,61,14,30,46,62,15,31,47
   ,63,16,32,48,64,4,19,34,49,8,23,38,53,12,27,42,57,16,31,46,61,13,26,39,52,13,25,37
3070  DATA 49,14,26,38,50,15,27,39,51,16,28,40,52,16,27,38,49,1,6,11,16,17,22,27,32,33,38,43,48,49,54,59,64,1,5,9,13,2,6,10,14,3,
   7,11,15,4,8,12,16,17,21,25,29,18,22,26,30,19,23,27,31,20,24,28,32,33,37,41,45,34
3080  DATA 38,42,46,35,39,43,47,36,40,44,48,49,53,57,61,50,54,58,62,51,55,59,63,52,56,60,64,4,7,10,13,20,23,26,29,36,39,42,45,52.
   55,58,61,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27
3090  DATA 28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64
3100  DATA 1,1,1,1,1,50,1,40,0,23264
3110  DATA 1,3,3,8,8,64,1,40,0,128,1
3120  DATA 1,1,5,50.1,30,350,1,40,00
3130  DATA 1,1,3,15.1,30,200,0,40,00
3140  DATA 1,2,2,15.1,10,90,0,40,0,3
3150  DATA 1,2,3,30.1,15,200,1,40,00
3160  DATA 1,2,4,10.1,20,150,0,8,6,5
3170  DATA 1,2,4,10.1,20,150,0,8,9,6
3180  DATA 1,2,3,15.1,15,100,1,12,10
3190  DATA 1,2,2,30.1,15,200,1,40,00
3200  DATA 0,1,5,2,10,15
3210  DATA 64,0,130,0,133,0,139,0,146,0,160,0,175,0,188,0,202,0,208,0,216,0,224,0,238,0,247,0,251,0,0,1,3,1,10,1,25,1,33,1,46,1,5
   7,1,68,1,80,1,92,1,101,1,113,1,124,1,129,1,134,1,143,1,151,1,160,1,170,1,184,1,197,1,210,1,221,1,233,1,245,1,255
3220  DATA 1,10,2,23,2,32,2,40,2,53,2,62,2,74,2,86,2,98,2,108,2,121,2,134,2,146,2,154,2,165,2,176,2,189,2,201,2,210,2,222,2,235,2
   ,243,2,0,3,7,3,73,73,0,9,33,32,36,44,0,64,64,36,13,54,7,0,64,37,60,12,108,190,45,30,23,45,30,254,36,0,9,41,56,63
3230  DATA 12,105,28,63,28,12,37,141,55,59,0,73,41,60,223,12,12,12,12,220,59,46,5,0,8,36,32,12,21,190,21,13,30,31,42,13,4,0,8,8,8
   ,32,60,0,73,28,28,36,12,12,5,0,73,12,12,36,28,28,7,0,8,12,12,60,28,76,110,30,22,55,13,222,7,0,72,33,31,40,37,172
3240  DATA 42,6,0,9,12,36,0,64,40,45,45,0,73,4,0,8,12,12,12,12,4,0,8,36,36,12,45,21,62,23,23,22,45,12,36,4,0,9,45,28,36,36,188,6,
   0,41,45,229,219,12,12,37,33,28,63,23,6,0,8,14,45,12,228,103,12,60,63,55,0,73,33,44,28,36,188,30,30,46,45,0,8,21
3250  DATA 45,12,36,28,63,39,44,45,53,0,64,118,45,12,228,63,39,12,12,45,6,0,9,36,12,12,12,60,63,55,0,8,172,42,101,228,63,28.100.4
   5,14,54,0,41,101,12,36,228,63,23,118,45,5,0,64,9,4,32,0,9,12,36,32,0,73,225,28,28,12,12,12,6,0,8,40,45,37,56,63
3260  DATA 55,0,9,12,12,12,28,28,28,6,0,9,33,32,12,12,28,63,23,5,0,8,36,36,12,45,21,62,51,46,44,150,63,63,0,33,36,44,40,168,14,54
   ,54,4,60,63,7,0,33,36,36,44,45,14,246,63,150,45,12,36,0,8,36,36,12,45,21,150,186,63,7,0,33,36,36,44,45,14,54,54
3270  DATA 23,63,7,0,33,36,36,44,45,181,26,63,150,45,37,0,33,36,36,44,45,181,26,63,7,0,9,45,37,60,215,35,36,100,45,53,0,33,36,36,
   108,9,54,54,54,199,24,40,45,0,9,45,28,36,36,7,40,53,0,8,14,45,12,36,36,60,0,33,36,36,108,9,30,30,30,14,14,14,4,0
3280  DATA 33,36,36,172,146,18,45,37,0,33,36,36,172,86,100,12,54,54,54,7,0,33,36,36,172,114,14,86,36,36,36,7,0,8,36,36,12,45,21,5
   4,54,30,63,7,0,33,36,36,44,45,21,54,59,63,0,8,36,36,12,45,21,54,254,14,21,63,63,0,33,36,36,44,45,21,246,63,14,14
3290  DATA 21,7,0,8,14,45,12,228,63,28,100,45,21,6,0,73,36,36,252,40,45,53,0,8,36,36,108,9,54,54,54,59,63,0,64,36,36,77,49,54,246
   ,30,28,6,0,33,36,36,108,9,54,254,190,13,12,54,7,0,33,12,12,28,28,108,9,246,22,21,62,0,73,36,228,28,108,9,246,7,0
3300  DATA 41,45,229,219,12,12,12,12,60,63,55,0,33,36,36,44,45,245,27,54,54,46,45,4,0,72,73,28,28,28,28,6,0,41,45,37,36,36,60,63,
   119,9,54,54,7,0,64,12,12,21,21,6,0,41,45,45,0,0,255,255,0,0,5,0,12,0,52,0,86,0,187,0,9,1,27,27,27,59,60,60,60,60
3310  DATA 45,45,45,45,45,45,45,45,53,53,53,53,53,53,53,53,61,63,63,63,63,63,63,63,60,60,60,60,4,0,63,63,60,60,45,53,37,53,
   37,53,46,36,53,46,52,53,61,63,63,63,63,53,45,45,45,45,53,61,63,63,63,63,4,0,31,31,28,28,13,21,5,40,42,40,50,33,40
3320  DATA 50,33,48,49,49,57,59,59,59,59,43,42,41,41,41,41,49,57,59,59,59,59,35,0,35,33,36,36,55,54,39,36,55,54,39,60,62,46,62,54
   ,21,46,5,44,54,55,22,46,46,44,46,36,21,46,44,4,32,63,63,63,37,12,45,42,37,45,4,12,36,39,37,39,55,62,36,60,54,62
3330  DATA 36,60,54,54,46,42,42,55,6,0,35,33,36,36,55,54,39,36,55,54,39,60,62,46,62,54,21,46,37,53,62,54,50,53,37,53,37,44,50,37,
   37,32,63,63,63,37,12,45,42,37,45,36,33,60,44,60,60,54,39,36,55,54,39,36,55,54,54,21,21,61,54,0,27,3,40,9,9,9,6,0
3340  DATA 27,3,40,9,9,9,57,0,27,3,40,9,9,9,6,0,255
3350  DATA 16,80,8,80,8,100,7,68,16,80,4,100,8,121,6,121,6,130,6,121,12,100,3,80,8,100,7,80,8,68,8,80,16
3360  DATA 41,80,1,80,1,80,2,80,1,80,1,80,2,80,1,80,1,130,2,148,2,163,2,80,1,80,1,80,2,80,1,80,1,80,2,130,1,163,1,148,2,121,2,80,
   2,80,1,80,1,80,2,80,1,80,1,80,2,80,1,80,1,130,2,148,2,162,2,80,1,163,1,182,5,163,1,148,1,130,2,163,2,130,4
3370  DATA 30,121,3,121,3,130,3,148,3,148,3,130,3,121,3,100,3,80,4,80,4,100,3,121,3,121,5,100,2,100,7,121,3,121,3,130,3,148,3,148
   ,3,130,3,121,3,100,3,80,4,80,4,100,3,121,3,100,5,80,2,80,24
3380  DATA 16,163,9,100,3,121,3,130,3,148,3,163,9,130,3,163,11,130,3,163,11,100,3,130,3,100,3,48,6,130,3,100,15
3390  DATA 55,80,4,100,3,121,24,121,1,100,2,121,3,130,3,148,6,130,3,121,3,80,20,48,6,130,3,121,3,121,12,130,6,182,3,192,3,202,28,
   202,2,192,2,202,3,208,4,215,8,208,4,215,4,223,24,163,3,208,4,202,3,192,28,182,3,182,3,176,3,148,3,130,3,121,3,80,4
3400  DATA 148,1,130,3,121,1,80,2,19,6,48,3,80,2,130,1,121,1,80,4,130,1,121,1,80,4,48,6,19,12,68,4,80,16
3410  DATA 22,163,9,182,4,182,15,163,9,148,4,130,15,148,9,163,4,182,9,163,4,148,20,163,9,182,4,182,15,163,9,148,4,130,15,148,9,16
   3,4,148,9,130,4,130,18
```

VIC 20 遊戲 介紹

MASTERMIND



在過往的MASTER－MIND的程式有很多。但估顏色的不多（亦可估數字的），可有雙色亦很少。

此程式是爲VIC－20寫的，此機種比較新，所以在很多雜誌都沒有VIC的程式。希望各VIC的機主，如有程式都能公諸同好。

此程式是有二級的：（1）單色；（2）雙色，亦可三色或四色，但在句48－56的副程式產生三色或四色的機會很低。故衹作兩級的玩法。此程式在運行時佔5K的記憶，所以要加上－3K的擴展。如嫌長的話可刪去800－2030的MASTER－MIND大字；歌仔亦何改短一些，程式中的歌仔是小蜜蜂。

程式的結構：

15　開始時的畫面

16　將歌仔的DATA放入記憶

22－39　玩法的指示

48－56　產生亂數及謎底

130　輸入的答案的數字

131及137　產生“嗶”一聲（不同的）

135－136　取消輸入的答案和消除畫面的字。

141　延時

142　當估中時的畫面及歌仔

150　清書答案下面的字

160－162　將畫面停下

200－300　答案的輸入及顏色的印出

310－350　輸入的答案是否可以CHECK

505　印出估了的次數

510－525　檢查對色的對位置並印出線索

526　答案全對時的處理

530－570　檢查對色的字並印出線索

800－2030　開始時MASTER－MIND大字的DATA

2050－2053　歌仔的DATA

F（90）　歌仔的DATA

FA（4）　答案答對時用

FB（4）　謎底被估中時用

BX（12），MX（12）產生亂數

LP$　垂直位置

NP$　水平位置

CR$　顏色

QX　級數

CM　估了的次數

L　垂直位置的計算

N　水平位置的計算

Q　印出對色對位置時的符號位置計算

R　印出對色的符號位置計算

A$　估的數字或顏色

AT$　四個估的字或顏色（答案）

GT$　謎底

AS$　問輸入的是否答案

C　計算畫面不要到底

L，N，開始時畫面的等號的位置

GH$　符號

WT　延長所需時間

```
137 POKES+1,224:WT=5:GOSUB141:POKES+1,0:RETURN

1 REM ****************
2 REM *  MASTER MIND *
3 REM *BY CHU SUN CHI*
4 REM * 5TH JUNE,1982*
5 REM ****************
12 DIM F(90),FA(4),FB(4),BX(12),MX(12)
13 LP$="
14 NP$="
15 PRINT"":S=36874:POKES+4,15:GOSUB799
16 FORI=1TO62:READF(I):NEXT
22 CR$="":POKES+5,27
23 PRINT"YOU HAVE SIX COLOR TO CHOICE (1--6)":PRINT:PRINT:PRINT
24 FORI=1TO6:PRINTMID$(CR$,I,1);"        IS ";I:NEXT:GOSUB160
25 PRINT"   THERE ARE TWO RATE (1 AND 2)":PRINT
26 PRINT"(1) NO DOUBLE COLOR"
27 PRINT"(2) DOUBLR COLOR"
28 PRINT:PRINT"  NOW ENTER YOUR RATE":PRINT
29 INPUTQX:IFQX<1ORQX>2THEN10:GOSUB131
31 PRINT"   AFTER YOU ENTER THE ANS."
32 PRINT"   A STATEMENT IS THE ANS. READY Y/N WILLAPPEARE"
33 PRINT"IF YOU ENTER (N) THE ANS. WILL DISAPPEARE AND YOU MUST ENTER AGAIN"
34 PRINT"IF YOU ENTER (Y) THEN THE ANS. WILL BE CHECKED BY ME":GOSUB160
35 PRINT"   OTHERWAY, IF YOU CANNOT GIVE THE ANS."
36 PRINT"   YOU CAN TPYE ANS . THEN THE ANS. WILL BE PRINTOUT":GOSUB160
37 PRINT"   AFTER THE ANS. IS CHECKED BY ME, I WILL GIVE YOU HITS"
38 PRINT"   IS MEAN ONE OF THE COLOR IS RIGHT"
39 PRINT"  * IS MEAN ONE OF COLOR AND POSITION IS RIGHT":GOSUB160
48 I1=0:FORQR=0TO6*(QX-1)STEP6:FORI=1TO6
50 MX(I+QR)=0:BX(I+QR)=I*10:NEXTI,QR
52 FORI=6*QXTO1STEP-1:I1=INT(RND(1)*I)+1
54 MX(I)=BX(I1):BX(I1)=BX(I):NEXT
56 FORI=1TO4:G$=STR$(MX(I)/10):GT$=GT$+G$:NEXT
120 GOTO180
130 PRINTLEFT$(LP$,L);LEFT$(NP$,N);:N=N+2:AT$=AT$+A$
131 POKES+2,224:WT=5:GOSUB141:POKES+2,0:RETURN
135 PRINTLEFT$(LP$,L)"                    ":PRINTLEFT$(LP$,L+2)"
     "
136 PRINT"       ":N=2:AT$=""
137 POKES+1,224:WT=5:GOSUB141:POKES+1,0:RETURN
141 FORX=1TOWT*30:NEXT:RETURN
142 PRINTLEFT$(LP$,L+2);"  YOU ARE WIN !! ":GT$=""
143 FORI=1TO62:POKES+2,F(I):FORK=1TO150:NEXTK
144 POKES+2,0:FORK=1TO30:NEXTK
145 NEXTI:GOSUB160:RETURN
150 PRINTLEFT$(LP$,L+2)"                    ":PRINT"          ":RETURN
160 PRINTLEFT$(LP$,22)" PRESS SPACE BAR TO        CONTINUE"
161 GETSB$:IFSB$<>" "THEN161
162 GOSUB137:RETURN
180 CM=0:L=5:C=1
185 POKES+5,205:PRINT""
186 PRINT"                 "
187 PRINT"  1  2  3  4  5  6"
190 N=2:Q=17:R=20:AT$=""
200 FORI=1TO4
210 GETA$:IFA$=""THEN210
211 IFVAL(A$)<1ORVAL(A$)>6THEN210
220 GOSUB130:PRINTMID$(CR$,VAL(A$),1)"  "
300 NEXT
301 PRINTLEFT$(LP$,L)""AT$
310 PRINTLEFT$(LP$,L+2)"IS THE ANS. READY Y/N"
320 INPUTAS$
325 IFAS$="ANS"THENPRINTLEFT$(LP$,L+5);GT$:STOP
330 IFAS$="N"THENGOSUB135:GOTO200
340 IFAS$="Y"ORAS$=""THENGOSUB150:GOTO500
500 REM** CHECK ANS. AND GIVE HITS
503 FORI=0TO4:FA(I)=0:FB(I)=0:NEXT
505 CM=CM+1:PRINTLEFT$(LP$,L);"";CM
510 FORI=1TO4
520 IFMID$(GT$,I*2,1)=MID$(AT$,I,1)THENPRINTLEFT$(LP$,L)LEFT$(NP$,Q);"*":GOTO52
3
522 GOTO525
```

```
523 Q=Q+1:FA(I)=I:FB(I)=I:GOSUB137
525 NEXT
526 IFQ=R+1THENWT=10:GOSUB141:GOSUB142:POKES+2,0:GOTO22
530 FORJ=1TO4
535 FORI=1TO4
540 IFFB(J)=JTHEN570
545 IFJ=IORFA(I)=ITHEN560
550 IFMID$(GT$,J*2,1)=MID$(AT$,I,1)THENFA(I)=I:GOTO555
552 GOTO560
555 PRINTLEFT$(LP$,L)LEFT$(NP$,R);"▚▇":R=R-1:GOSUB137:GOTO570
560 NEXTI
570 NEXTJ:L=L+2
571 PRINTLEFT$(LP$,L-1)
575 C=C+1:IFC=7THENPRINT:PRINT:PRINT:PRINT:PRINT:PRINT:PRINT:C=6:L=15
578 GOTO190
799 REM** MASTER MIND
800 POKES+5,8:POKES,198
810 READL1,N1,GH$
811 IFL1=0ANDN1=0THENPOKES,0:GOSUB2010:GOTO800
812 IFL1=0ANDN1=1THENPOKES,0:GOSUB2020:RETURN
819 PRINT"▚"
820 PRINTLEFT$(LP$,L1)LEFT$(NP$,N1)GH$
830 GOTO810
901 DATA15,0,"▮",16,0,"▚ ▇"
902 DATA17,0,"▚▄▇",18,0,"▮"
903 DATA19,0,"▮",20,0,"▮"
904 DATA21,0,"▮"
905 DATA16,1,"▖",17,1,"▚ ▇"
906 DATA18,1,"▚▇▇",16,2,"▚▛▇",17,2,"▚▄▇"
909 DATA15,3,"▮",16,3,"▮",17,3,"▮",18,3,"▮"
910 DATA19,3,"▮",20,3,"▮",21,3,"▮"
911 REM** CHARACTER A
913 DATA15,5,"▚▇▇",16,5,"▚▄▇",17,5,"▮",18,4,"▚▇▇",19,4,"▚▇▇",20,4,"▮",21,4,"▮"
915 DATA16,6,"▮",17,6,"▮",18,6,"▚▇▇",19,6,"▚▛▇",20,6,"▘",20,7,"▮",21,7,"▮"
916 DATA19,5,"▄",20,5,"▚▄▇"
921 REM** CHARACTER S
923 DATA 16,10,"▚▄▇",15,10,"▖",15,9,"▚ ▇"
924 DATA15,8," ▖",16,8,"▚▄▇",17,8,"▚▀▇",18,8," ▘",18,9,"▚ ▇",18,10,"▖"
925 DATA19,10,"▚▄▇",20,10,"▚▛▇",21,10,"▘",21,9,"▚ ▇",21,8," ▘",20,8,"▚▀▇"
931 REM** CHARACTER T
932 DATA15,11,"▚▇▇",15,12,"▚ ▇"
933 DATA15,13,"▚ ▮",15,14,"▮",16,12," ▌",16,13,"▌"
934 DATA17,12," ▌",17,13,"▌",18,12," ▌",18,13,"▌",19,12," ▌"
935 DATA19,13,"▌",20,12," ▌",20,13,"▌",21,12," ▌",21,13,"▌"
941 REM** CHARACTER E
942 DATA15,15,"▚ ▇",16,15,"▚ ▇",17,15,"▚ ▇",18,15,"▚ ▇",19,15,"▚ ▇"
943 DATA20,15,"▚ ▇",21,15,"▚ ▇",15,16,"▚ ▇",15,17,"▚ ▇"
945 DATA18,16,"▚ ▇",18,17,"▚ ▇"
946 DATA21,16,"▚ ▇",21,17,"▚ ▇"
951 REM** CHARACTER R
952 DATA15,18," ▌",15,19,"▚ ▇",16,18," ▌",16,19,"▮",17,18," ▌"
953 DATA17,19,"▚▀▇",18,18," ▌",18,19,"▚ ▇",19,18," ▌",19,19,"▚▄▇"
954 DATA20,18," ▌",20,19,"▮",21,18," ▌",21,19,"▮",15,20,"▚ ▇"
955 DATA15,21,"▚▀▇",16,21,"▚▀▇",17,21,"▚ ▇",18,21,"▘",17,20,"▄"
956 DATA18,20,"▚▄▇",19,20,"▚▀▇",20,20,"▚▄▇",20,21,"▖"
957 DATA21,20," ▘",21,21,"▚▀▇"
1000 DATA0,0,""
1001 DATA16,7,"▚         ▇"
1002 DATA17,7,"*",17,16,"*"
1003 DATA18,7,"*  MIND  *"
1004 DATA19,7,"*        *"
1005 DATA20,7,"▚         ▇"
2000 DATA0,1," "
2010 FORI=1TO6:PRINT:POKES+1,234
2015 WT=10:GOSUB141:POKES+1,0:NEXT
2020 FORI=1TO3:PRINT:POKES+1,240
2025 WT=10:GOSUB141:POKES+1,0:NEXT
2030 WT=100:GOSUB141:RETURN
2050 DATA215,207,207,0,209,201,201,0,195,201,207,209,215,215,215,0
2051 DATA215,207,207,0,209,201,201,0,195,207,215,215,195,0,0,0
2052 DATA201,201,201,201,201,207,209,0,207,207,207,207,207,209,215,0
2053 DATA215,207,207,0,209,201,201,0,195,207,215,215,195,0

READY.
```

GAMES介紹



HANG MAN

各位不同年齡的朋友，除非你小於十歲，否則應會玩過「估字遊戲」，我當年玩此遊戲之時，經常被考起，頂心頂肺之至，故作此程式去考人，看看別人被考起時的神態——好看過電影。

如果你不是虐待狂，那麼又可以讓別人出字給你去估，看看自己的能力如何。

玩法：——

在玩這遊戲前必要先INIT－IALIZE兩個TEXT F－ILE，INIT的方法可以參照LISTING2，首先把LI－STING2打入，然後放進一隻沒有WRITE－PROTEC－T的磁碟，這時便可以RUN了，當RUN完後，CATALOG就會發現多了兩個TEXT FI－LE，一個叫WORD FIL－E／L16，另一個叫HIGH－EST SCORE／L32，它們都是RANDOM ACCE－SS TEXT FILE，選擇RAT FILE的原因是容易EDIT，當要改變內容時，RA－T FILE比SQUENTI－AL TEXT FILE方便得多，但RAT FILE所佔的容量比較多，對大FILE來說，並不化算，但本遊戲的FILE不大，故可放心使用。

當CREATE了兩個TEX－T FILE後，便可以NEW去LISTING2，跟着打入L－ISTING1，再將LIST－ING1 SAVE入同一張磁碟上，便可以正式玩了。當然，要沒有KEY－IN ERROR才成，否則，請閣下慢慢DEBUG了。

在遊戲內有一份小小的MANUAL，可以給你4個選擇，第一：INSTRUCTION，第二：WORD FILE MAINTENANCE，第三：HIG－HEST SCORE TAB－LE，第四：START。

遊戲的玩法已在INSTRU－CTION OPTION之中，故此不在這裏多費筆墨了。第三個OPTION是讓你看一看十個最高分的LIST，同時也可以把你的成績給朋友看，揚威一番。S－TART的意思便即是STAR－T，大家應該瞭解，毋用多說。要費筆墨的便是WORD FIL－E MAINTENANCE

這一項3：當APPLE或你的朋友給字你估之前，都要先將那些字輸入WORD FILE，而第二個OPTION便是USER和WORD FILE接觸的BR－IDGE，在WORD FIL－E內，每十個字爲一組，當你I－NIT WORD FILE時，實際上已把一組字SAVE入，以後你還可以無限制的輸入，只要磁碟能容納就成，唯一的限制是每字不得長過15個CHARACT－ERS，這是因爲WORD F－ILE內每個RECORD的長度在OPEN時已DEFINE爲16CHARACTERS，正好容納15個INPUT CHAR－ACTERS和CARRIAGE RETURN（CHR$（13））。正因爲每組字都是十個，所以無論輸入或輸出，都以十個爲單位，如果在輸入時不是十個一組的話，COMPUTER便會跟你沒完沒了，直至它得到十個字爲止，你

還可以EDIT一組以前輸入的字，只要選擇該組然後由頭輸入十個字便成。

在DISK ACCESS時，千萬不要按CTRL－C或RESET，因爲好可能你會干擾了那兩個FILE的內容，令它們GONE WITH THE WIND！而只要DISK Ⅱ上IN USE燈不亮時，便可以隨時BREAK，絕不會影響TEXT FILE。

本程式並沒有經過COMPACT，所以可讀性比較高，至於流程圖，因爲太過簡單，無論你懂不懂BASIC都可以很容易便跟得到，不擬在此闡釋。

頭腦清醒的你會發現在程式內的三個大SUBROUTINE都可以用GOTO寫成而無須用GOSUB，我之所以把它們寫成SUBROUTINE是因爲要加上多些OPTION時容易很多，毋須左拆右拆，諸君以爲然否？

一般寫BASIC的朋友，如非寫UTILITY程式，都不大有機會使用TEXT FILE，而DOS3.3TEXT FILE的一些細節便可能不太大了，故此解釋本遊戲所用的兩個TEXT FILE。

第一個是WORD FILE／L16，L16是我慣用的IDENTIFIER，好等以後一看就知這是一個每個RECORD有16CHARACTERS長的RAT FILE，每當DECLARE RECORD LENGTH時，都要留一個CHARACTER給CHR$（13）（即RETURN），所以每一個WORD便最多只可以有15CHARACTERS。

在WORD FILE內，RECORDφ是用來儲藏字組的數目，如果有5組字，RECORDφ便會是5，餘此類推，而字是以十個爲一組，所以RECORD1至1φ是第一組字，RECORD11至2φ是第二組，SO ON SO FORTH。

第二個FILE是HIGHEST SCORE／L32，顧名思義，這FILE儲藏著最高分的玩家的姓名和分數，可以永留紀念，和WORD FILE不同的是：字FILE的長度不定，視乎字組多少，但HIGHEST SCORE FILE則永遠只有11個RECORD（Rφ至R1φ），而RECORDφ不用，所以HIGHEST SCORE FILE的長度永遠不變。這FILE的第一個CHARACTER是／（STROBE），這是因爲WRITE入TEXT FILE時，LEADING SPACE會被CUT去，如果你的名字前面有SPACE便不會被WRITE入，但加上一個NON－SPACE CHARACTER在RECORD前，便可以防止這種情形發生。

結語

本程式可以發展的地方很多，而程式的結構亦已預備了，方便大家各自發爲。

TABLE 1是變數表及它們出現的行數，各位改時應會方便很多。

希望各位虐待別人或被虐待時愉快。

布老圖 

```
]LIST

100 D$ =  CHR$ (4)
110  DATA APPLE,COMPUTER,LOGIC,CRAZY,RANDOM,EASY,FUNNY,
     UNIVERSITY,RESUME,RETURN
120  FOR I = 1 TO 10: READ WO$(I): NEXT
130  PRINT D$"OPEN WORD FILE/L16,L16"
140  FOR I = 1 TO 10
150  PRINT D$"WRITE WORD FILE/L16,R"I
160  PRINT WO$(I): NEXT
170  PRINT D$"WRITE WORD FILE/L16,R0"
180  PRINT 1
190  PRINT D$"CLOSE WORD FILE/L16"
200 DU$ = "/": FOR I = 1 TO 29:DU$ = DU$ + " ": NEXT :DU$ = "/"
210  PRINT D$"OPEN HIGHEST SCORE/L32,L32"
220  FOR I = 1 TO 10
230  PRINT D$"WRITE HIGHEST SCORE/L32,R"I
240  PRINT DU$: NEXT
250  PRINT D$"CLOSE HIGHEST SCORE/L32"
260  END
```

```
]PR#0
]&X

]POKE1657,136

]&X

AL$(  1400, 1750, 1830, 1840
BE    1940, 1950
BU    1140, 1150, 1230, 1250, 1270
BU$   1140
CH    1850, 1860
CO    1690, 1900, 1910
CW    1620, 1900, 2020, 2030
D$    1210, 1230, 1260, 1280, 1400, 1410, 1420, 1440, 1450, 1470, 1490, 1630, 1650, 1670, 2160, 2180, 2200
DA    1690, 1730, 1850, 1890, 2000
F     1680, 1690, 1710, 1850, 2000
GP    1060, 1140, 1250, 1270, 1430, 1500, 1580, 1590, 2060
GP(   1500, 1590, 1620
GU$   1830, 1850
HS    1500, 1700, 1920, 2040
HS$(  1320, 1480, 1500, 2090, 2120, 2130, 2140, 2150, 2190
HY    1730
I     1150, 1160, 1170, 1220, 1230, 1240, 1320, 1380, 1460, 1470, 1480, 1570, 1640, 1650, 1660, 1750, 1910, 1970, 2000, 2030, 2090, 2100
, 2120, 2140, 2170, 2180, 2190
J     1170, 1910, 1970, 2030, 2100, 2120, 2130, 2140, 2150
KZ    1380, 1770, 1780, 1820, 1910, 1970, 2030
MA    1740, 1910, 1920, 2030, 2040, 2090, 2130
NA$   1550, 1560, 1570, 1580, 1700, 2130
P     1590
RI    1690, 1850, 1860, 1890
S     1760, 1780, 1790, 1800, 1810, 1830, 1840
SE    1580, 1590, 1620, 1650, 1710, 2070
SE$   1580
TE    1170, 1850
TI    1940, 1960, 2000, 2030, 2060, 2110
WC    1620, 2060
WD$(  1160, 1170, 1240
WR    1690, 1860, 1870, 1880, 1900
W$(   1660, 1690, 1850, 2000
X     1690
YN    1520, 1530
YN$   1130, 1190, 1200, 1520, 2070, 2080, 2220, 2230


]LIST

1000  GOTO 1360
1010  HOME : VTAB 5: PRINT "THE COMPUTER GIVES YOU A CHOSEN WORD": PRINT "AND YOU ARE TOLD THE LENGTH OF THE WORD": PRINT "BY THE
    NUMBER OF THE HYPHENS"
1020  PRINT : PRINT "ALL YOU HAVE TO DO IS ENTRE AN ALPHABET": PRINT "EACH TIME YOU ARE RIGHT,": PRINT "APPLE WILL PUT IT ONTO TH
    E APPROPRIATE": PRINT "POSITION"
1030  PRINT : PRINT "IF YOUR GUESS IS INCORRECT": PRINT "IT JUST DISAPPEARS": PRINT "7 WRONG TRIALS WILL LEAD TO FAILURE"
1040  PRINT : PRINT "WORD SELECTING TECHNIQUE :": PRINT "PRESS '.' TO RIGHT, ',' TO LEFT": PRINT "PRESS THE SPACE BAR TO ENTRE WO
    RDS": PRINT "PRESS THE 'REPT' KEY FOR REPEAT ENTERING"
1050  VTAB 23: PRINT "PRESS ANY KEY TO CONTINUE...": GOSUB 1340
1060  HOME : VTAB 5: PRINT "THERE'RE "GP" GROUPS OF WORDS": PRINT : PRINT "EACH OF THEM CONTAINS 10 WORDS": PRINT "EACH WORD CARR
    IES 50 MARKS"
1070  PRINT : PRINT "YOU'LL RECIEVE THE MARK IN PROPORTION": PRINT "EACH WRONG GUESS WILL LOSE 5": PRINT "IF YOU FAIL TO GUESS TH
    E WORD,": PRINT "NO MARK WILL BE GIVEN"
1080  PRINT : PRINT "THERE'LL BE A BONUS UP TO 200": PRINT "WHICH IS GIVEN ACCORDING TO HOW MANY": PRINT "WORDS YOU CAN GUESS IN
    A GROUP"
1090  PRINT : FLASH : PRINT "ANYTIME YOU WANT TO QUIT, PRESS 'ESC'": NORMAL
1100  VTAB 23: HTAB 1: PRINT "PRESS ANY KEY TO CONTINUE...": GOSUB 1340
1110  RETURN
1120  END
1130  HOME : VTAB 8: HTAB 15: PRINT "PASS WORD ?": VTAB 10: HTAB 15: INPUT "";YN$: IF YN$ < > "HANG MAN" THEN  RETURN
1140  HOME : VTAB 8: PRINT "WHICH GROUP YOU WANT TO DEAL WITH ?": VTAB 10: INPUT "";BU$:BU =  INT ( VAL (BU$)): IF BU < 1 OR BU >
    GP + 1 THEN  RETURN
1150  HOME : VTAB 6: PRINT "GROUP "BU: FOR I = 1 TO 10
1160  VTAB 7 + I: HTAB 1: CALL  - 958: PRINT "ENTRE WORD # "I;: INPUT " ";WD$(I): IF WD$(I) = "" OR  LEN (WD$(I)) > 15 THEN 1160
1170  FOR J = 1 TO  LEN (WD$(I)):TEST =  ASC ( MID$ (WD$(I),J,1)): IF TEST < 65 OR TEST > 90 THEN 1160
1180  NEXT : NEXT
1190  VTAB 21: HTAB 1: PRINT "ARE THEY CORRECT ? (Y/N) ";: GET YN$: IF YN$ = "N" THEN 1150
1200  IF YN$ <  > "Y" THEN 1190
1210  PRINT : PRINT D$"OPEN WORD FILE/L16,L16"
1220  FOR I = 1 TO 10
1230  PRINT D$"WRITE WORD FILE/L16,R"(I + BU * 10 - 10)
1240  PRINT WD$(I): NEXT
1250  IF BU <  = GP THEN 1280
1260  PRINT D$"WRITE WORD FILE/L16,R0"
1270  PRINT BU:GP = BU
1280  PRINT D$"CLOSE WORD FILE/L16"
1290  RETURN
1300  END
1310  HOME : VTAB 6: PRINT " RUN UP  PLAYERS";: HTAB 29: PRINT "SCORES": PRINT " ======  =======";: HTAB 29: PRINT "======"
1320  FOR I = 1 TO 10: VTAB 8 + I: HTAB 5 - (I = 10): PRINT I"     " MID$ (HS$(I),2,28): NEXT
1330  VTAB 23: HTAB 4: PRINT "PRESS ANY KEY TO CONTINUE......"
1340  POKE  - 16368,0: WAIT  - 16384,128: POKE  - 16368,0: RETURN
1350  END
1360  DATA 173,48,192,136,208,4,198,7,240,8,202,208,246,166,6,76,0,3,96
1370  CLEAR : TEXT : HOME
1380  FOR I = 768 TO 786: READ KZ: POKE I,KZ: NEXT
1390  VTAB 1: HTAB 1: INVERSE : PRINT " HANG MAN  - BY PLUTO NG -  AUGUST 1981 ": NORMAL : POKE 34,1
1400  DIM AL$(26):D$ =  CHR$ (4)
1410  PRINT : PRINT D$"OPEN WORD FILE/L16,L16"
1420  PRINT D$"READ WORD FILE/L16,R0"
1430  INPUT GP
1440  PRINT D$"CLOSE WORD FILE/L16"
1450  PRINT : PRINT D$"OPEN HIGHEST SCORE/L32,L32"
1460  FOR I = 1 TO 10
1470  PRINT D$"READ HIGHEST SCORE/L32,R"I
1480  INPUT HS$(I): NEXT
1490  PRINT D$"CLOSE HIGHEST SCORE/L32"
1500  DIM GP(GP):HS =  VAL ( MID$ (HS$(1),21,9))
1510  HOME : VTAB 9: HTAB 5: PRINT "<1> INSTRUCTIONS": PRINT : HTAB 5: PRINT "<2> WORD FILE MAINTAINANCE": PRINT : HTAB 5: PRINT
    "<3> HIGHEST SCORE TABLE": PRINT : HTAB 5: PRINT "<4> START"
1520  VTAB 17: HTAB 5: PRINT "WHICH ";: GET YN$:YN =  VAL (YN$): IF YN < 1 OR YN > 4 THEN 1510
1530  ON YN GOSUB 1010,1130,1310,1540: GOTO 1510
1540  POP
1550  HOME : VTAB 9: HTAB 8: PRINT "PLEASE GIVE ME YOUR NAME": PRINT : HTAB 8: PRINT "( 16 CHARACTERS MAX.)": PRINT : HTAB 8: INPUT
    "";NAME$: IF  LEN (NAME$) > 16 THEN NAME$ =  LEFT$ (NAME$,16)
1560  IF NAME$ = "" THEN 1550
1570  IF  LEN (NAME$) < 16 THEN  FOR I = 1 TO 16 -  LEN (NAME$):NAME$ = NAME$ + " ": NEXT
1580  HOME : VTAB 10: PRINT NAME$: VTAB 13: INPUT "WHICH GROUP WOULD YOU LIKE TO TRY ? ";SE$:SE =  VAL (SE$): IF SE < 1 OR SE > G
    P THEN 1580
1590 SE =  INT (SE): FOR P = 1 TO GP: IF GP(P) = SE THEN 1580
1600  NEXT
1610  VTAB 13: HTAB 1: PRINT "GET YOUR MIND SET AND --- GOOD LUCK !!!"
1620 WC = WC + 1:GP(WC) = SE:CW = 0
1630  PRINT : PRINT D$"OPEN WORD FILE/L16,L16"
1640  FOR I = 1 TO 10
1650  PRINT D$"READ WORD FILE/L16,R"(I + SE * 10 - 10)
1660  INPUT W$(I): NEXT
1670  PRINT D$"CLOSE WORD FILE/L16"
1680  FOR F = 1 TO 10
1690 RIG = 0:WRO = 0:COUNT = 0:DA =  LEN (W$(F)):X =  FRE (0): HOME
1700  VTAB 4: PRINT "HIGHEST SCORE EVER MADE : ";: INVERSE : PRINT " "HS" ": NORMAL : VTAB 7: PRINT "PLAYER :  "NAME$
1710  VTAB 10: PRINT "GROUP   :  "SE: VTAB 13: PRINT "WORD    :  "F
1720  VTAB 16: PRINT "GUESS   :": VTAB 20: PRINT "SELECT :"
1730  VTAB 17: HTAB 11: FOR HYPHEN = 1 TO DA: PRINT "-";: NEXT HYPHEN: PRINT
1740  VTAB 23: PRINT "SCORE   :  "MARK;: HTAB 23: PRINT "COUNT DOWN : 7"
1750  FOR I = 1 TO 26:AL$(I) =  CHR$ (I + 64): VTAB 20: HTAB 10 + I: PRINT AL$(I): NEXT
1760  VTAB 21: HTAB 11 + S: PRINT " "
1770  IF KZ = 155 THEN 2090
1780 S = S + (KZ = 174) - (KZ = 172)
1790  IF S < 0 THEN S = 25
1800  IF S > 25 THEN S = 0
1810  VTAB 21: HTAB 11 + S: PRINT "-"
1820  WAIT  - 16384,128:KZ =  PEEK ( - 16384): POKE  - 16368,0: IF KZ <  > 160 THEN 1760
1830 GUESS$ = AL$(S + 1): IF GUESS$ = " " THEN 1820
1840  VTAB 20: HTAB 11 + S: PRINT " ":AL$(S + 1) = " "
1850 CHECK = RIG: FOR TEST = 1 TO DA: IF GUESS$ =  MID$ (W$(F),TEST,1) THEN  VTAB 16: HTAB TEST + 10: PRINT GUESS$:RIG = RIG + 1
1860  NEXT : IF CHECK = RIG THEN WRO = WRO + 1
1870  VTAB 23: HTAB 36: PRINT 7 - WRO
1880  IF WRO > 6 THEN 1970
1890  IF DA <  > RIG THEN 1760
1900 COUNT = 50 - WRO * 5:CW = CW + 1
1910  FOR I = 1 TO COUNT: FOR J = 1 TO 5:KZ =  PEEK ( - 16336): NEXT :MARK = MARK + 1: VTAB 23: HTAB 11: PRINT MARK: NEXT
1920  IF HS < MARK THEN  VTAB 7: HTAB 28: INVERSE : PRINT MARK" ": NORMAL :HS = MARK
1930  VTAB 10: HTAB 21: FLASH : PRINT " YOU WIN !!! ": NORMAL
1940  FOR BELL = 1 TO 15: FOR TIME = 1 TO 150: NEXT
1950  VTAB 12: HTAB BELL + 20: PRINT  MID$ ("CONGRATULATIONS",BELL,1) CHR$ (7): NEXT BELL
1960  FOR TIME = 1 TO 2000: NEXT : NEXT : GOTO 2020
1970  FOR I = 1 TO 4: CALL 768: FOR J = 1 TO 65:KZ =  PEEK ( - 16336): NEXT : NEXT : CALL 768
1980  VTAB 10: HTAB 21: PRINT "OH ! WHAT A PITY !!": VTAB 12: HTAB 21: FLASH : PRINT " YOU'VE MISS IT !! ": NORMAL
1990  VTAB 16: HTAB 30: PRINT "<-- ANSWER"
2000  FOR I = 1 TO DA: VTAB 16: HTAB 10 + I: PRINT  MID$ (W$(F),I,1) CHR$ (7): FOR TIME = 1 TO 500: NEXT : NEXT
2010  GOTO 1960
2020 CW = 25 * 2 ^  INT ((CW - 4) / 2): IF CW < 25 THEN 2040
2030  FOR I = 1 TO CW STEP 5:MARK = MARK + 5: FOR J = 1 TO 5:KZ =  PEEK ( - 16336): NEXT : FOR TIME = 1 TO 200: NEXT : VTAB 23: HTAB
    11: PRINT MARK: NEXT
2040  IF HS < MARK THEN  VTAB 7: HTAB 28: INVERSE : PRINT MARK" ": NORMAL :HS = MARK
2050  VTAB 16: HTAB 30: PRINT "           "
2060  IF WC = GP THEN  VTAB 10: HTAB 21: PRINT "ALL GROUPS ARE DONE": VTAB 12: HTAB 21: PRINT "JUST WAIT A MINUTE": FOR TIME = 1 TO
    5000: NEXT : GOTO 2090
2070  VTAB 10: HTAB 21: PRINT "GROUP "SE" IS DONE    ": VTAB 12: HTAB 21: PRINT "CONTINUE ? (Y/N)  ";: GET YN$: IF YN$ = "Y" THEN
    1580
2080  IF YN$ <  > "N" THEN 2070
2090  FOR I = 1 TO 10: IF MARK <  =  VAL ( MID$ (HS$(I),21,9)) THEN  NEXT : GOTO 2220
2100 J = I: HOME : VTAB 8: PRINT "HALLELUJAH !!! IT'S A GREAT PLAY !": PRINT : FLASH : PRINT " YOU'RE ONE OF THE TOP TENS ! ": NORMAL

2110  VTAB 14: HTAB 1: PRINT "YOUR NAME WILL BE PUT": PRINT : PRINT "INTO THE HIGHEST SCORE TABLE": FOR TIME = 1 TO 5000: NEXT
2120  IF J < 10 THEN  FOR I = 9 TO J STEP  - 1:HS$(I + 1) = HS$(I): NEXT
2130 HS$(J) = "/" + NAME$ + " * " +  STR$ (MARK)
2140  IF  LEN (HS$(J)) < 29 THEN  FOR I = 1 TO 29 -  LEN (HS$(J)):HS$(J) = HS$(J) + " ": NEXT
2150 HS$(J) = HS$(J) + "/"
2160  PRINT : PRINT D$"OPEN HIGHEST SCORE/L32,L32"
2170  FOR I = 1 TO 10
2180  PRINT D$"WRITE HIGHEST SCORE/L32,R"I
2190  PRINT HS$(I): NEXT
2200  PRINT D$"CLOSE HIGHEST SCORE/L32"
2210  GOSUB 1310
2220  HOME : VTAB 10: PRINT "ANYBODY ELSE WANT A GAME ? (Y/N) ";: GET YN$: IF YN$ = "Y" THEN  RUN
2230  IF YN$ <  > "N" THEN 2220
2240  TEXT : HOME : VTAB 22: END
65534  REM PLUTO NG
65535  REM A.D.1982
```

Z-80 CPU卡制作

本文所介紹的Z80－A CPU卡是十分容易製作的。從圖一，我們可以知道，主要的部份只需要一些很簡單的IC便可以了。記憶體是由4K所組成的。第一組（0000……0FFF）的位置是在CPU卡內而且包括2KEPROM（0000……07FF）和2K RAM（0800……0FFF）。這卡板上的設計還應用了一新的多變化聲音合成器。在應用方面因爲只需要1K RAM便可，所以IC18和IC19（見圖一）是可以除去的。

在CPU板上，我 是沒有一定需要把記憶IC放置在相對的地址解碼器上的。大量的軟件最好的是貯存在一塊分離的EPROM板上。

緩衝器（BUFFER）

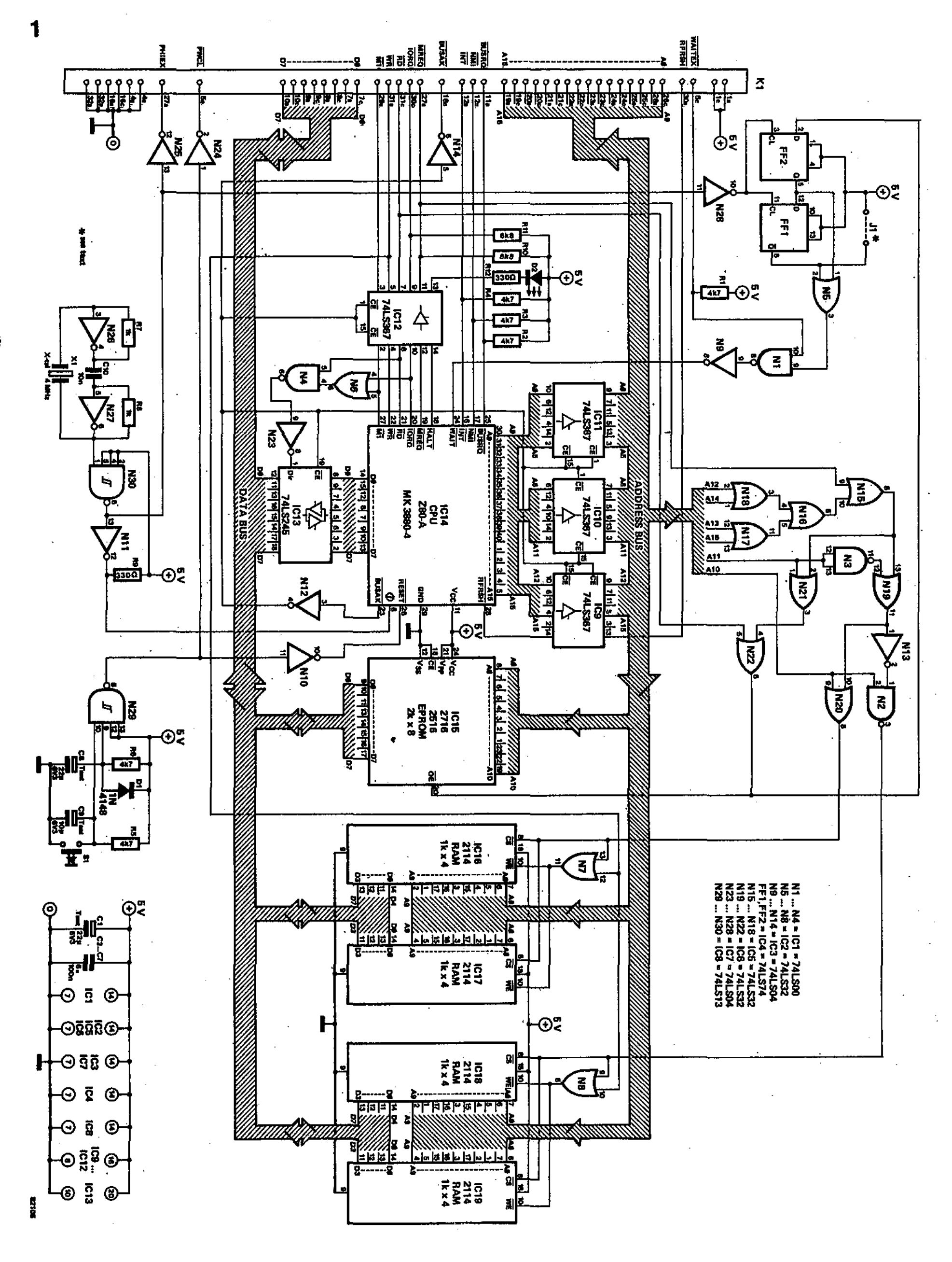
一塊設計良好的CPU板必定需要緩衝級。因爲CPU的輸出是不可能完全正接控制整個系統的。因應用在這裏（IC9……IC13）的緩衝器是三態（TRI－STATE），由BUSAK訊號，DMA或多重處理（MULTIPROCESSING）所啓動；Z80的設備仍然保留。

週期處理問題

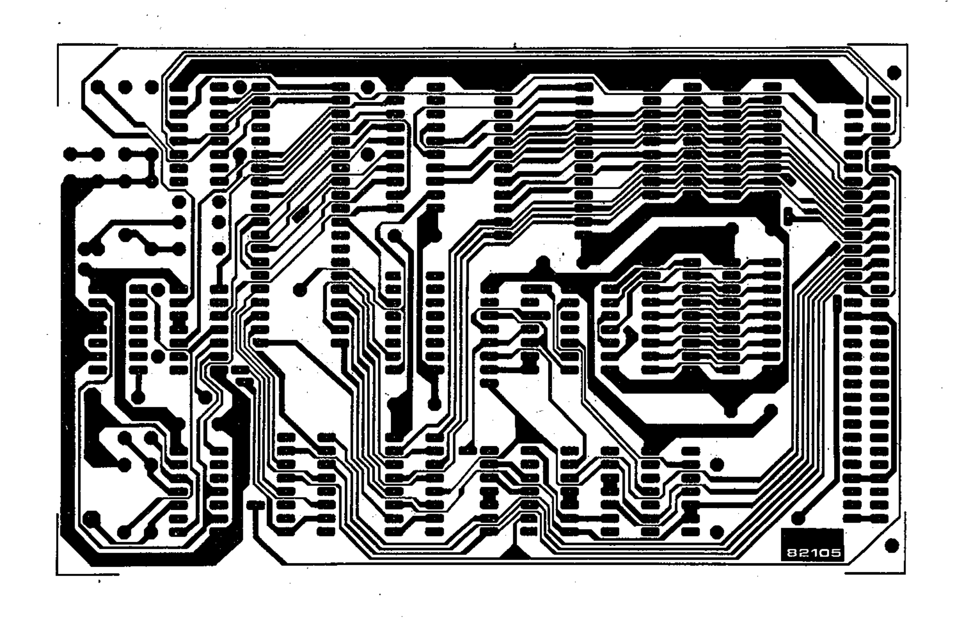
處理器是由一個4MHz 的晶體振盪器所驅動。爲Z80或MK3880－4CPU的最高時鐘頻率。一個標準的Z80或MK3880的時鐘頻率是不可能超過2．5MHz 的。在這處，筆者要說明多峰段（POLYFORMANT）是需要一個Z80－A（MK3800－4）的CPU。

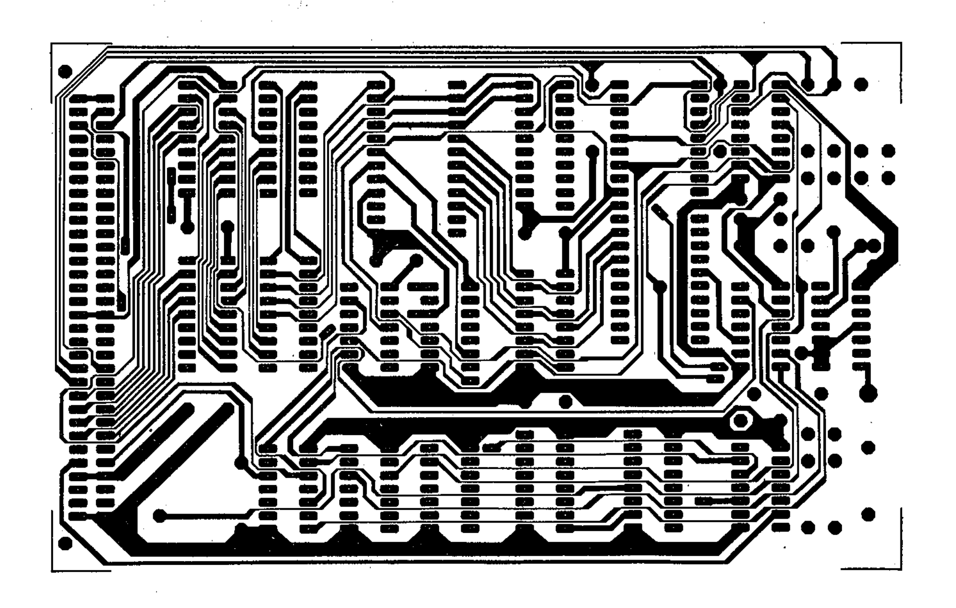
很明顯地，處理器的操作速度主要在決定執行一個程式所需要的時間。在POLYFORMANT中，CPU必定要測覽鍵盤，而在擴展的版本中，還要測覽所有的週邊設備。再者，還要將相關的數據輸送到POLYFORMANT的模件中。

這些處理所需的時間要視乎其反應速度。這些對鍵盤測覽（SCANNED）時尤其特別重要。測覽的速度越快，VCO便能夠以快速的時間對按入的鍵作出反應。利用POLYFORMANT所發展的軟件，一個VCO可以在2或3微秒內（MILLISECOND）。但是這個延遲的時間實在太細了，用家是很難發現的。



圖(一)：Z80A CPU板的電路圖。





圖（二）：雙面印刷電路板的銅軌圖樣。

處理

利用高時鐘頻率能夠自動呼喚對應的處理速度或提取時間。提取一個標準27·6EPROM（1015）的時間通常會比較長，而不能由CPU找到地址。數據輸入的時候，寫入RAM（IC16……IC1A）的時間便更加受到限制。要解決這個難題有兩個方法。第一個方法利用高速的記憶體，即是指EPROM和RAM的提取速度分別是350NS和250NS。後者在市面上是比較容易找到，但350NS的EPROM則比較困難找到了。嚴格來說，甚至350NS。

雖然另一個捷徑是植置OE（OUTPUT ENABLE）輸入代替CE（CHIP ENABLE）輸入。這樣一枚350NS的2716在使用時就不須附加任何額外的測量器具。

另一個方法就是減低CPU的速度而應用正常低速的EPROM。這種進行是加上等待週期（WAIT CYCLE）去讀取操作。一個週期維持的 間完全等於一個時鐘段落（CLOCK HERIOD），即250NS。所以增加一個單獨的等待週期將會延長EPROM提取時間至500NS。這種延遲是受FF1和FF2此兩個觸發器（FLIPFLOP）所影響。

這些觸發器（FLIPFLOP）只是在EPROM（IC15）被尋址的時候才合活躍的（FF2的輸入是低時）。只果一個350NS的EPROM存在，它們可以略去不用。當接上電線時，必須接上J1以代替IC4。這樣便減低了延遲電路（DELAY CIRCUIT）的活躍性。當測驗CPU時，筆者忠告讀者們首先嘗試第一個辦法和包括一個等待週期，以便確定一個慢度的EPROM將不會使整套系統過於複雜。

任何的外展記憶或週邊設備均能產生等待週期（藉着輸往WAITEX）

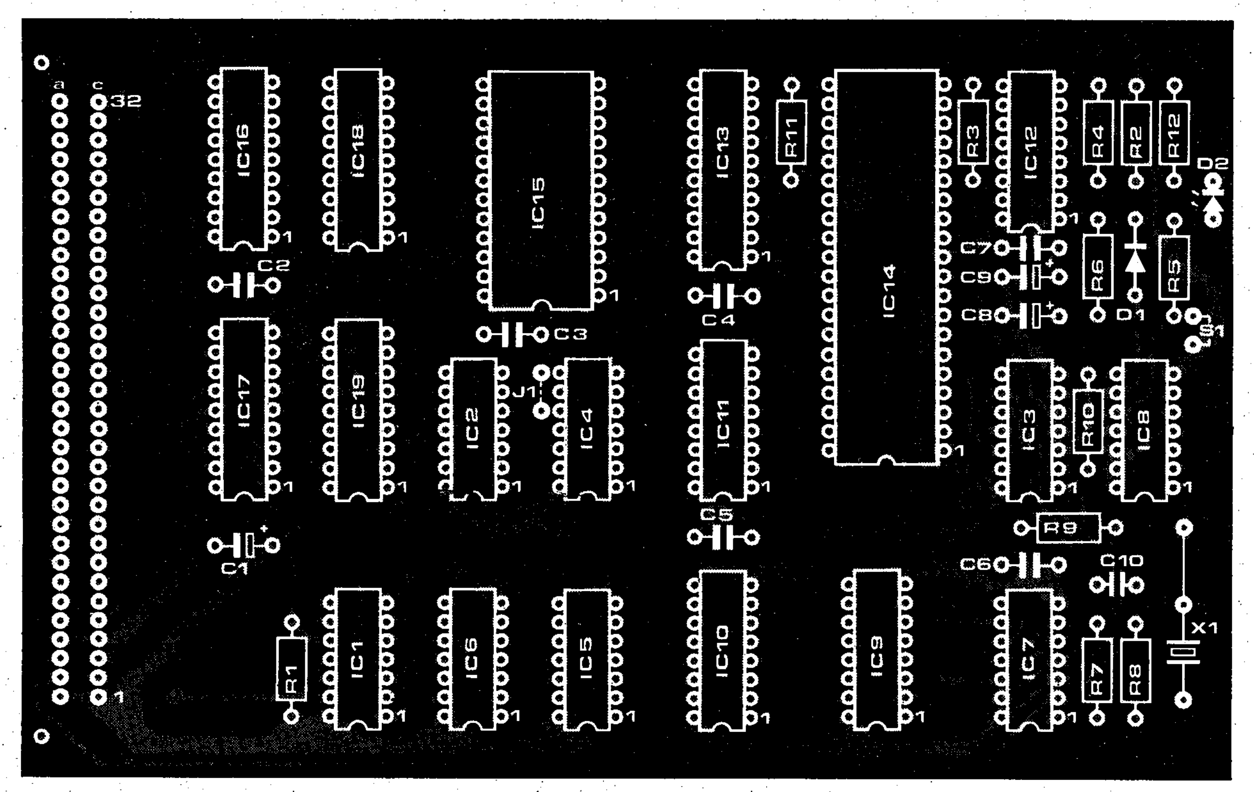
還原（Reset）

還原電路是用來啓動CPU的。當電源接通的時候R6、C8和D1便會從N9和N10載着CPU的低還原輸入一般時候。這是DWCL的訊號和用來還原任何其他接駁系統巴士線板的。一個外展的還原設備是緊急時用。

筆者忠告製作者將S1設置在比較困難接觸的地方，以免意外的觸摸此掣引致珍貴的資料失去。

印出的電路板

除3S1，所有在圖一的部件（COMPONENT）都是設置在一塊EUROCARD大細的，雙面的、印刷電路板內。（見圖2）



圖（三）：在ＣＰＵ板上的零件位置。

由於６４脚管連接器的脚管排列與ＥＬＥＫＴＯＲ巴士線的脚管排列相同；因此這電路板可與其他一些卡一起使用。

當把部件放置在ＣＰＵ卡的時候，必須特別小心，因爲在板上的一些地方，銅軌的互相距離十分細小，銲接的時候可能使電路〝ＳＨＯＲＴ〞了。雖然這塊板已提供了一個銲接保護罩，但小心還是很重要。

■■

零件表

電阻

Ｒ１……Ｒ６＝４Ｋ７

Ｒ７、Ｒ８＝１Ｋ

Ｒ９＝３３０Ω

Ｒ１０、Ｒ１１＝６Ｋ８

Ｒ１２＝４７０Ω

電容器

Ｃ１、Ｃ８＝２２μ／６・３Ｖ

Ｃ２……Ｃ７＝１００Ｎ陶瓷電容

Ｃ９＝１０μ／６・３Ｖ

Ｃ１０＝１０Ｎ　ＭＫＨ

半導體

Ｄ１＝１Ｎ４１４８

Ｄ２＝ＬＥＤ

ＩＣ１＝７４ＬＳ００

ＩＣ２、ＩＣ５、ＩＣ６＝７５ＬＳ３２

ＩＣ３、ＩＣ７＝７４ＬＳ０４

ＩＣ４＝７４ＬＳ７４

ＩＣ８＝７４ＬＳ１３

ＩＣ９…ＩＣ１２＝７４ＬＳ３６７

ＩＣ１３＝７４ＬＳ２４５

ＩＣ１４＝Ｚ８０Ａ（ＯＲ　ＭＫ３８８０－４）

ＩＣ１５＝２７１６／２５１６

ＩＣ１６…ＩＣ１９＝２１１４－Ｚ　ＯＲ　２１１４Ａ－５（提取時間２５０ＮＳ）

其他

Ｓ１＝按鈕開關

Ｘ１＝４ＭＨz 晶體

Ｋ１＝６４－ＰＩＮ連接器（ＣＯＮＮＥＣＴＯＲ）

6502改6809方法

MOTOROLA經過不斷的努力，最近又推出一款新的產品—6809CPU。這款產品，主要是從6502改良得來。故此，採用者只需將目前的6502除下，再插上新的6809便可。可以說，6809是6502的"新一代超級產品"。事實上，使用6809除部份之硬件要有些少更改外，其它重要地方，完全毋須額外修改，便可擁有一部更爲快捷、有效及實用的電腦。

在發展了高速電腦系統之後，MOTOROLA又已推出另一擁有16BIT 結構的微型訊息處理器。6809稱爲"超級的6502"，其中的一個原因是6809的寄存器名稱，與6502完全一樣。此外，兩個系統的特性，亦頗爲相似。當然，MOTOROLA這塊晶片較之6502其工作速度更爲快捷及有效。至於兩者的構造分別，可參看圖（一）。

從圖中所見，6809擁有額外的8BIT累加器（REGISTER）及一可變、直接頁區寄存器。另一方面6502CPU，只有一個"零頁寄存器"。實際上，現時的6809有256個頁區可用。6809還有一個好處，它有兩個累加器；A及B。這兩個累加器，亦可組合爲一個16BIT 累加器。指示裝置方面，與6502操作器頗爲相似，其實，經改變的地方很少，祇有記憶及地址的形態方面。

分支指令特別有效，微訊息處理器可分支在—16…+15，—128…+127，或—32768……+32767的地址範圍。新的指示，如BRA（BRANCH ALWAYS（經常分支））及BSR（BRANCH TO SUBROUTINE（分支至子程序））等，可使程序存儲在記憶的任何區域，而不需倚賴絕對地址及改變字節（BYTE）。這等程序叫做"可重新配置"程序。事實上，新的系統介紹了一個新的地址形態—"相對程序寄存器"形態（"PROGRAM COUNTER RELATIVE"MODE）。無可否認這形態極端有力，它保證任何記憶位址在任何時刻，視程序寄存器（PROGRAM COUNTER）的內容來編址。

可能，每樣事情，總有利有弊。如採用6809，慣用6502的讀者，就要放棄他們其中的一個喜愛的地址形態—間接索引形態。雖然，6809不能將間接索引地址形態，但它的其它優點，足可彌補這缺點。

在6809中，它的地址編目程序形式，與6502略有不同。運算碼（OPCODE）包括有一個單獨字節及隨後的連絡字節（POST BYTE）。這連絡字節可包括有5—BIT變位。每對字節代表了8—BIT或16—BIT變位。有效的地址計算方法，是將目錄（INDEX）及變位（DISPLACEMENT）相加：即有效地址（EFFECTIVE ADDRESS）=目錄（X、Y、S、U、A、B或C寄存器的內容）+變位。

如果變位被安排於—16…+15範圍內，目錄地址形態的指示只會包括兩個字節：運算碼及連絡字節。

雖然，6809沒有眞正的間接索引尋址，記憶亦可間接地利用索引地址形態存取。這時，指針指示有效地址的ADH所貯存的記憶位置。至於ADL，它貯存在下列記憶位置：在6809CPU中，ADH及ADL經常是處於順序。但在6502中，次序剛好相反（ADL，ADH）。無可否認，間接的設備特別有用，因它保證陣列及符號表可用高級程序語言列出。

另外，累加器可用作索引寄存器（INDEX REGISTERS）。這句話的意思表示累加器不但可用作增量及減量，而且更可用於數學或波爾代數（BOOLEAN ALGEBRA）中。故此，指數亦能計算。這過程可稱爲累加器編目形態。還有，6809CPU擁有兩枝疊層指針，S及U。S爲一枝16—BIT信息保持指針，其功用與6502的一枝指針相同。所有由子程序及機械寄存器的回返地址，均自動存貯於S這個寄存器的臨時保持信息部份。同時，這指針還用作執行一切的中止控制信號。

至於U指針，它純粹是程序員的助手。它本身亦是16—BIT闊及主要用作輸入緩衝和於原文編輯時，用作迴路指針。

至現在爲止，相信讀者們都會認爲6809及6502系統是擁有甚爲相似的程序結構。雖然，它們的地址形態差不多完全相同，但明顯地，兩者不同之處，是6809 CPU提供更爲有力的指示裝置，其速度又比6502爲快。所以，以事論事，擁有6502，就要把它追上時代，改爲6809。不過，改換6809前，下列事情要清楚—

（1）硬件需要作些微修改。

（2）適合6809的軟件，較6502爲多。

（3）6809系統可用BASIC，FORTRAN，PASCAL及CROSS ASSEMBLER等語言的磁碟。CROSS ASSEMBLY是"雙方向"性的，它的範圍包括由6809至Z—80或Z—80至6809。

（4）所有6809系統，只有一種標準的軟碟控制形式。但6502則有多種控制形式。

本文寫到至此，應說明如何將6502系統改爲6809電腦。首先，將6809CPU與一個4MHz石英晶體及兩個於電路板上的電容接連。然後整個元件裝於一個40—管

脚DIL接連器。裝好後，只需將6809代替6502便可。管脚安排可見圖（二）

改裝過程：

（1）由插製中除去6502CPU。

（2）插入6809的新組合。

（3）將6502的操作系統（貯存於ROM或EPROM），用6809的代替。這程序可用ASSIST 09監視程序。

此外，6809亦有原文編輯（TEXTEDITOR），連接程序（LINKER）及磁碟操作系統（DOS）等附件。因此，讀者已可將電腦運用自如。

6502

寄存器	6809	6502
X－寄存器	16位	8位
Y－寄存器	16位	8位
疊層指針	16位	8位
A累加器	8位	8位
直接頁區寄存器	可變	只限零頁及固定
狀態寄存器	8位	7位
程序計數器	16位	16位

6809

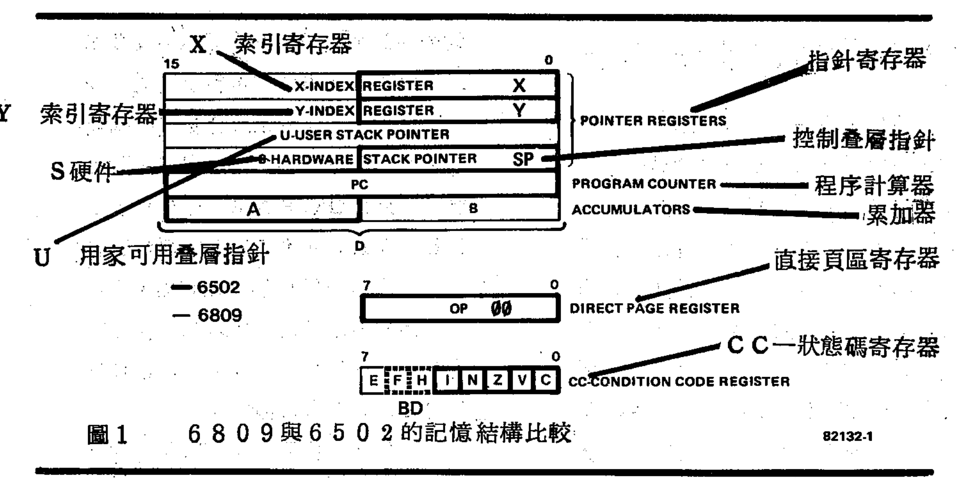
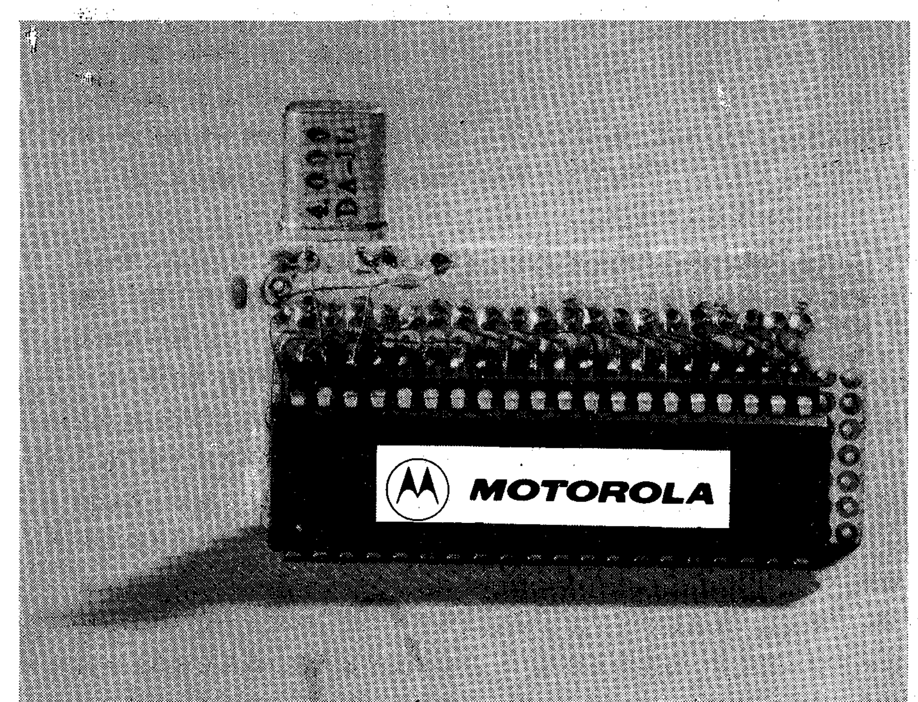


圖1　6809與6502的記憶結構比較



利用孖仔方式，同將6809新底板插回原6502插座上。

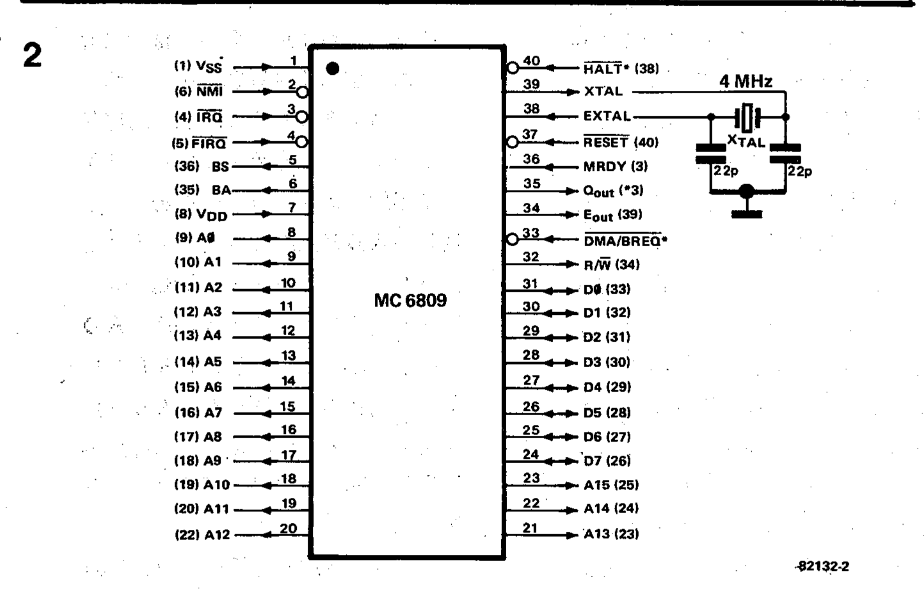
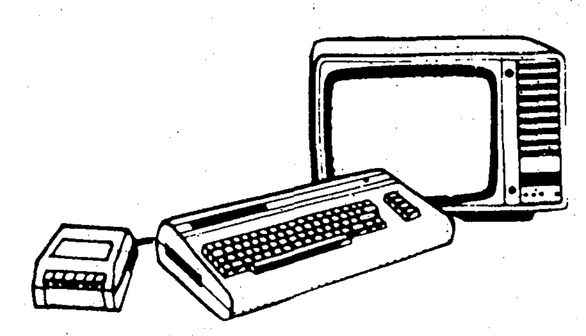


圖2　6809的針脚排位圖，括號中的號碼相對6502的針脚號碼

VIC-20 CASSETTES

卡式機



分析

卡式操作方法

ＶＩＣ可以外接一部卡式錄音器用來存放或提出資料。錄音機用６條綫接往ＶＩＣ—寫入、讀出、摩打感應及另兩條電源綫，地與＋５Ｖ。綫路包配圖如圖１。

卡式是經兩個ＶＩＡ由Ｉ／Ｏ綫所控制，圖２表示由ＶＩＡ所引出的控制卡式機的各條電綫接法。

由介面ＶＩＡ所引出控制摩打的部份是先經三個晶體管接駁成推動電路。故此摩打可以直接驅動。輸往摩打的是未爲穩壓的＋９Ｖ，電流量是５００ＭＡ。在第一個６５２２上將ＣＡ２綫反覆觸發便可打開或關掉卡式錄音機。

將摩打開動的方法：

POKE 37148,PEEK(37148) AND 241 OR 14

把摩打關掉方法：

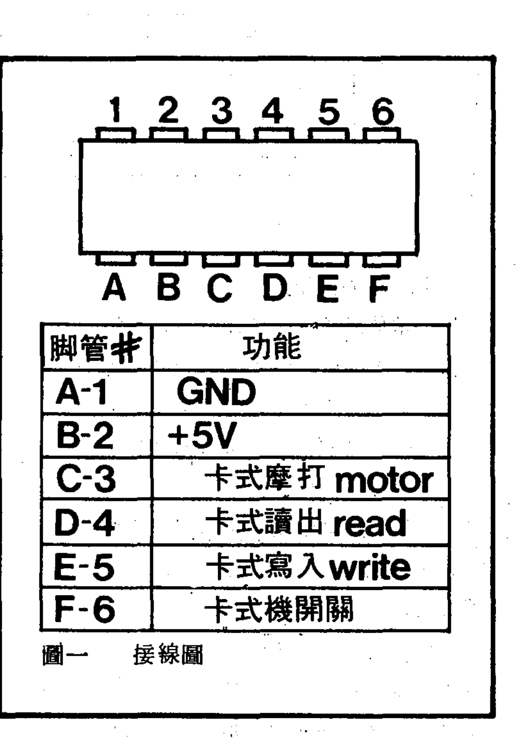
POKE 37148,PEEK(37148) OR 12 AND NOT 2

在ＶＩＡ＃１上的ＰＡ６綫是感應輸入綫，它是接往卡式錄音機上一個開關，用來檢測錄音機的按鍵是在什麼狀態，是播放，回捲或快速中那一鍵被按下。這個鍵的作用是用來檢測在運行讀入或寫入錄音帶程式時，播放鍵有無按下。

在需要按播放鍵時，如果不慎按下回捲或快速前進鍵的時候，系統是無法測出這個錯誤，而會一直以爲已經按下了播放鍵。因爲當按下播放鍵後錄音便會進行而感應開關亦會關掉，故此是在錄音時最好先按下錄音鍵。

卡式讀入綫是經接往ＶＩＡ＃２的ＣＡ１而寫入、綫則接往ＶＩＡ＃２的ＰＢ３。當進行讀入過程，操作系統利用ＣＡ１的中斷旗號去檢查卡式讀入綫有無變動情況出現。讀入與寫出均由操作系統控制，唯一需要的硬件是訊號放大及波形整形綫路。

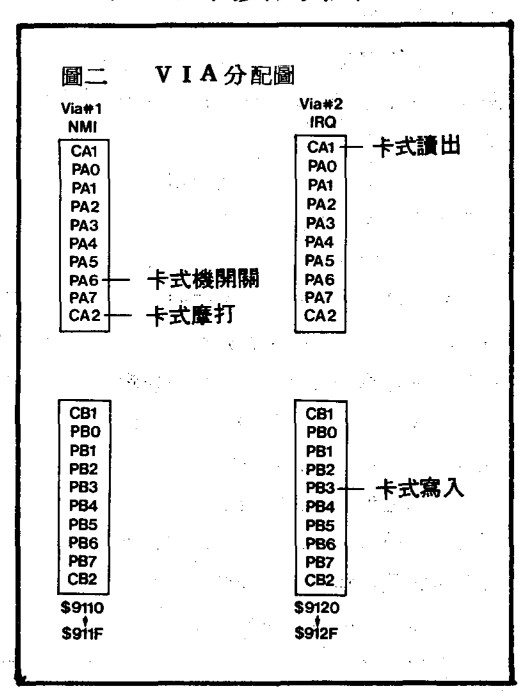
這個綫路全部在一塊印刷電路板上，藏在卡式座的裡面。功用是提供與錄音頭一個準確的電壓及電流，與及將讀入磁頭的電流加以放大。將會產生一個５Ｖ方波輸出，足夠在ＣＡ１或ＣＢ１綫上產生中斷。



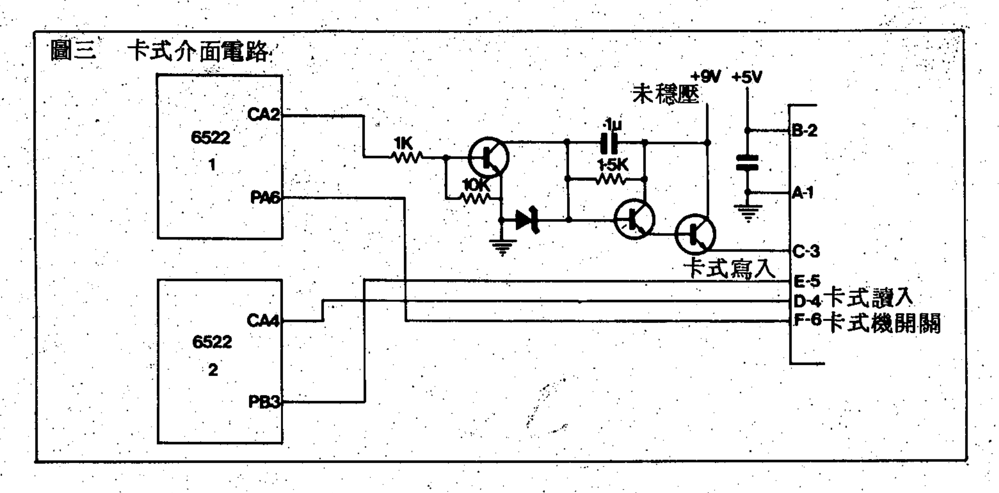
脚管#	功能
A-1	GND
B-2	+5V
C-3	卡式摩打 motor
D-4	卡式讀出 read
E-5	卡式寫入 write
F-6	卡式機開關

圖一　接線圖

卡式外接方法



圖二　ＶＩＡ分配圖



圖三　卡式介面電路

卡式機的操作方法

正常使用下卡式機是定爲元件號碼1。正在使用中的I／O元件號是存在地址186，連同這個號碼，加上邏輯檔案號碼，與及第二地址等是用在卡式機的資料檔案存入及提取。

由於第二地址是決定卡式的操作狀態，故此是很重要數據，使用中的第二地址是存在185的位置中，正常起始值是零。如果第二地址是零，卡式機是在讀入狀態。如果定爲1則是寫入，是2的時候則是用來寫入及在檔案完結時加上一個"END OF TAPE"的標記。

邏輯檔案號碼是在1與255之間，是準許在同一個元件中存放多重檔案。對於卡式機來說用途不大，一般是用在磁碟上。在卡式機使用時，邏輯號與元件號相同，而它是存放在184位置中。

VIC操作系統準許在卡式機中存放兩種不同的檔案，程式檔案種數據檔案。這兩個名稱的分界是很模糊的；程式是可以存在數據檔案，而數據反之亦可以存在程式檔案中。

這兩種檔案在使用時分別很少，但眞正的分別是在記憶體中的內容。與其叫它們做程式及數據檔案，不如直接叫做二進制數據種ASC11數檔案更爲合適。

二進制數據是一般用來存放程式的，因爲它是由操作系統來完成，將一個區域中的內容存起來，而這個區域由一個開始地址及至完結地址來劃定。稱它做二進制數據檔案的原因是存放在卡式帶的是在指定區域中每一個記憶地址裡的數據的二進制數值。

BASIC句是利用TOKEN存入記憶中，TOKEN是利用一個數作爲代表一個BASIC語句，例如用1代表REM，2代表PRINT等。好處是可以節省記憶位置。故此使用TOKEN時，BASIC語句的排列便不可像在鍵盤一一顯示看見到的一樣。反之，它們是用部份編碼方式存在記憶中。部份編碼方式的好處是在使用二進制檔案時可以更容易及更快而效力更高。在存放或提取機器碼程式時二進制數檔案亦最主要的方法。

二進制數檔案的開始地址是存放在172及173中。存放程式將數據放入這個地址。連同一般設定爲零與4的存放開始的記憶位置，便可以指向BASIC文件區的1024位置。

利用存放程式可以將這個位置改變指向記憶中任何一點。記憶區的地址是存放在174及175中。通常在存放一個BASIC程式時，是定在"雙零"句的地址（DOUBLE－ZERO BYTE），這是用來終止連接地址用。故此完結地址亦可以需求更改。

更改了這些地址後便不可以利用正常的SAVE程式了，因爲這個程式會自動將開始及完結地址給予一個初始值。故此要另外再寫一段機器碼初始化程式溶合所需的操作系統副程式。在DEFAULT時，一個SAVE句會寫入一個二進制數據檔案而一個LOAD句會讀入一個二進制數據檔案。

ASCⅡ檔案一般用來存放數據但亦可以用來存放程式。存放的規格是同鍵盤畫面所見的一樣。ASCⅡ檔案可以由BASIC直接制造和讀入。而二進制檔案是直接用機器語制造和讀入的，雖然仍然也可以在程式中使用LOAD和SAVE。

ASCⅡ檔案必須要用OPEN語句打開，定出邏輯檔案，元件號碼，第二地址及檔案名稱。操作系統將這些參數翻譯後便準許用家存放或提取在指定元件的檔案。

數據要經一個特別元件寫入一個ASCⅡ檔案中，用語句PRINT入一個指定的邏輯檔案號碼，再由邏輯檔案命令READ將數據讀入。

二進制檔案將連續記憶位置中的內容讀入，但ASCⅡ則載入一串變數。存取ASCⅡ檔案需要將卡式機不斷開關，每一次保存幾句數據。VIC解決這個難題的方法是利用一組緩衝器，一組192句磁帶緩衝器，所有要寫入或讀出的數據先放入這個緩衝器。當緩衝器滿載時卡式機才開動。

在卡式帶上的數據儲存方式是每組192句，而摩打是在這些組句中的空隙開動，這個空隙是2秒鐘，足夠摩打加速和減速了。第一個192字的緩衝寄存器是放在地址178與179中。緩衝寄存器中文字的數目是存在位置的166中。

設計員可以利用這些位置去控制在數據檔案中所剩餘的空間。假如，卡式中開了一個檔案，再執行了POKE166， 191，那麼卡式帶緩衝寄存器中所有內容——無論是空的——都會載入卡式帶中。假如所有記錄

是用191句或其倍數存在卡式帶時，適當利用上面的命令，可以將紀錄中的部份空置或全部空置，那麼，在將來便可放入適當的數據了。

無論檔案是用二進制或ASCⅡ方式寫入，基本的錄入的方法仍然是一樣的。所採用的編碼方式一貫的COMMODORE獨特方法，這個方法據稱無論寫入或提出數據時的可靠性很高。程式或數據中每一句是經操作系統譯成三種不同音調脈冲，就是：頻率1,488HZ的長脈冲，1,953HZ 的中脈冲與及2,840HZ 的短脈冲。

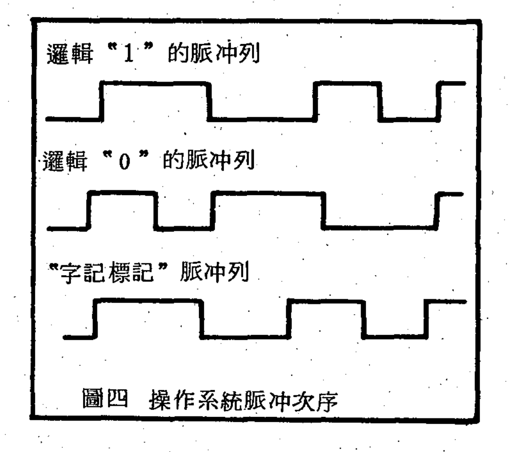
所有的脈冲均是方波，佔空比是1：1。中脈冲每一個週期是256μS 高電平狀態，256μS 低電平狀態。

操作系統用9MS去紀錄包括8個數據BIT，字標記，及奇偶BIT所組成的一句。數據BIT由一串0與1組成，分別是中程脈冲及短程脈冲。"1"是由一個中脈冲及跟隨一個短脈冲。而"0"是短脈冲跟隨一個中脈冲。每一個BIT包括兩個方波，一個長及一個短，總共需時864μS，從圖4中可以看出波形的定時。

奇偶檢錯碼BIT是用來檢查有無錯誤，編碼好當8個數據BIT一一用一個長及短脈冲。狀態由8個數據BIT所決定。字句標記將每一句數據隔離及發消息至操作系統，現在是另一個新句的時候了！字句標記編碼方法是一個長脈冲跟隨一個中脈冲，見圖4。

每一句數據用8.96MS存放，192句的一組ASCⅡ數據便要用1.7秒。但實際的錄取發覺是用上5一7秒。原因有兩個；由於恐怕錄音帶的磁粉會剝落，每一組192句都是重覆錄下兩次；第二個理由是組（192句）與組之間有2秒鐘的間隔。

廣泛應用誤差檢測這個方法是VIC的錄音帶系統較別家更為可靠的原因，有兩級檢查錯誤的方法，第一



圖四 操作系統脈冲次序

是將數據組成8BYTE一組，而計算出第9BYTE，是CHECKSUM數。CHECK—SUM是將8個數據BYTE加在一起後其最小有限值。

讀帶時，如果8BYTE中的一個BIT產生錯誤，例如"1"變為"0"或反轉，那麼，CHECKSUM便可以檢查出這個錯誤。為了要做到這一點，要將這8句加在一起，計算出的數據如果與檢查數（CHECKSUM）有分別，便是有ERROR了。

但檢測出ERROR不一定有好處，必需有一個方法防止出錯，VIC－20是每一組數據（192句）均錄下兩次，錄兩次的好處是，如果一次錯，第二次對，查出的方法是利用CHECKSUM，則可以利用第二次將第一次糾正。而錄兩次的另一個好處是可以驗校VERIFY，將兩組數據互相比較。這可以測出那些經CHECKSUM亦找不出的錯誤。

採用三種不同的脈冲，是有別於FSK（頻率漂移輸入法），優點是操作系統很容易測出速度的變化。通常FSK方式是利用硬件方式去查出及校正頻率的漂移，但VIC－20則採用軟件方式。

每一段數據寫入前都有10秒帶頭標記。帶頭標記有兩個功能。第一是讓摩打運轉到正確速度。第二，READ程序利用帶頭標記中的短脈冲來調整所有TIMING，以便同錄音帶的脈冲得到同步。

因此操作系統可以利用一個校正系數去對付變化很多的錄音帶轉速而不致影响讀入的效果。整個系統的定時必須要準確，而VIC便是採用石英振蕩產生時鐘脈冲，與及利用VIA＃2計時器1與2。每個紀錄間空隙僅在ASCⅡ檔案時採用，作用是讓摩打關掉後減速，或打開後加速，有了這個間隙，存放或提取每一個數據就更為可靠。

每個紀錄RECORD間的間隙約為2秒長，由一串短脈冲組成，正如10秒的帶頭標記一樣。每一組間仍然有空隙。每一組192句紀錄後便是組區完結記號，包括一個長脈冲及50個短脈冲。其後是另一個192句／組的紀錄開始。

無論ASCⅡ或二進制檔案都將第一個紀錄共192字作為檔案標記區。這個標記區包含檔案名稱，開始記憶地址，結尾地址。而在ASCⅡ檔案則錄音帶緩衝寄存器的開始及結尾地址。在二進制檔案中則是指向程式所存的記憶區位置。

檔案名稱長度可至128句，檔案名稱的長度是記在位置183中，在LOAD或OPEN命令中讀入的要與所需要的FILENAME比較中，名稱相同則操作系統會讀入檔案，如果不相同，則會找尋第二個的10秒間隙，再讀入另一個HEADER比較。

FILENAME是在讀或寫操作中存放於開始地址187及188的記憶區中。當操作完畢後，會重置在操作系統之另外一個位置。開始位置一般是定在用家可用記憶區的開始地址。

開始地址放在172及173中。結尾地址則放在174及175。通常結尾地址程式佔用的最高句記憶，但亦可以更改的，只要開始地址高便可以了。

RS-232C卡式介面制作

現時，使用卡式錄音機來作私人電腦的資料紀錄介面，已開始廣泛流行起來。可是，所採用的介面未必能適合不同的資料形式。有見及此，筆者特別譔寫本文，介紹一個簡單錄音機介面的製作。而此製作，主要是採用ＲＳ２３２Ｃ標準，以便能適合大部份儀器使用。

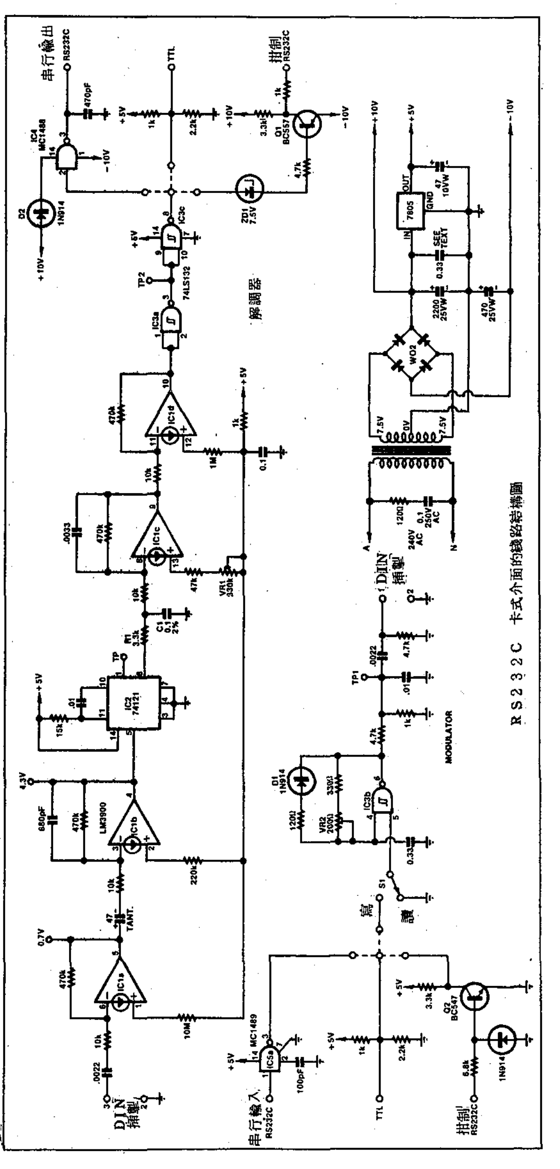
雖然，很多讀者都使用卡式錄音機來儲存其私人電腦的數字信息。但是，這些錄音機，很少符合ＲＳ２３２Ｃ標準，通常，它們處理串行資料的速度，只有３００波德（Ｂａｕｄ）。（註：波德是信號發送速度單位及發極速率單位。假如資料處理速度爲１２００波德的話，則表示此介面的紀錄工作及程序、資料檔案和串行資料再生速度較３００波德快４倍。）

如果錄音機能擁有一個ＲＳ２３２Ｃ介面：其功用會大爲提高。這時，由一個電腦存入錄音帶資料，可以在另一電腦讀出，而毋須局限於只能使用同一種類電腦，才能重讀此錄音帶內的資料。不過，它們需要有一共通硬件，這硬件是擁有ＲＳ２３２Ｃ介面的串行資料埠，而其亦需擁有同一波德速度的工作能力。

可能，部份讀者會擁有一個能通過ＲＳ２３２Ｃ介面來推動某一高品質及昂貴印字機的私人電腦。而另一方面你的朋友擁有這個特別的印字機的話，要將資料印出，就會是輕而易擧的事了。此時你所要做的事，只需將卡式機與印字機接連一起，再輸入程序運行，便能將存於錄音帶上的所需資料，全部印出。

另外的一個典形例子，如果使用者希望將由程序產生的資料輸送，但恰巧數據通信儀器已被其他人仕使用。遇此情形，程序可照常運行，而其輸出資料可輸送入卡式機儲存，當數據通信儀器可以使用時，才由卡式機輸出所存資料。

通常，任何配備ＲＳ２３２Ｃ介面的儀器，均能利用一個ＵＡＲＴ或ＵＳＡＲＴ來將平行數據轉爲串行數據（或串行數據轉爲平行數據）。在接收資料數據後，儀器能悉別串行數據，較少的非累積時間誤差，在一般情況是會消去的。


ＲＳ２３２Ｃ　卡式介面的總線路構圖

本文介紹的製作，儘量要降低元件成本，及將線路簡化，同時亦顧及元件的供應。原本，這製作的設計，是考慮及採用ＦＳＫ調制及相位鎖定迴路解調器。但後來這概念決定放棄，因現時的１００％振幅調制方法，雖然仍擁有甚差的噪音抗性，可是，其線路較爲簡單及實際。

調制器（Ｔｈｅ　ｍｏｄｕｌａｔｏｒ）

調制器是非常簡單的。它採用一個Ｓｃｈｍｉｔｔ　２輸入"與非門"（ＩＣｂ）觸發器來作爲一個選通電路振盪器。在這段電路中，有兩點要特別留意：第一點是它提供了同步功能；在此，輸入數字波形的正轉換及它產生的載波之第一個轉換會重合。第二點是Ｄ１及電阻１２０Ω被加進電路，以增進及改良載波波形的對稱。

圖（一）所示，爲調制器之波形及時間。由於輸入部份的負轉換缺乏同步，故此，於再生情況下，會造成與載波周期成正比例的時間誤差。所以，載波頻率愈大，則誤差會愈小。

本製作的錄音機能有效處理的最大載波頻率爲６．１ＫＨｚ。一個操作於７．５ＫＨｚ的樣辦，如與一高品質錄音機一起使用，它的處理速度，可達ｚ４００波德。不過，調制器輸出，可以適當調節，來匹配不同的錄音機。

調制器輸入端的讀一寫掣，於讀資料時，會使載波失去功能，以避免錄音帶進入的低電平載流子，與其產生可能性的干擾。

解調器（Ｔｈｅ　ｄｅｍｏｄｕｌａｔｏｒ）

由錄音帶送出的訊號，會被ＩＣ

1a及IC1b（屬於LM3900四路運算放大器的一部份）放大。至於輸入耦合電容值（0·0022uF），是經試驗後決定，這電容數值是能過濾DC偏壓錄音機內的固有低頻噪音。放大器IC1a和IC1b是DC偏移，故此，只有輸入訊號的負段才會被放大。當載波爲零時而噪音出現，放大器亦不會工作。IC1b的輸出，將被輸入擁有脉冲闊度大於載波的半個周期之單穩態集成電路，IC2的觸發器輸入端。另外，集成網路是接於單穩態集成電路的管脚6（輸出Q）。圖（二）所示爲此部份電路的波形。

電路的操作全倚賴於一個事實；電容的充電時間是較放電時間爲長。所以，最後的結果是帶正電荷。這電荷被IC1C及IC1d放大後，轉變爲TTL電平。這過程由Schmitt觸發器"NAND門"（IC3a所完成。此外，外加門（IC3）將結果轉爲正確的羅輯形態。

放大器IC1c擁有偏移調整，它可以用來調校再生數字波形的正確時間關係。這個解調器有一好處；任何資料的單載波周期的損失，均不會產生誤差。至於使用AGC電路，相信是不甚需要的。因爲，以現時之放大器增益，輸入幅度甚至波動達80%，亦沒有產生任何差錯。

錄音機部份

在這個製作所採用的錄音機，是菲利蒲卡式錄音機模型N2238。選擇這型號的錄音機之原因，是其價錢相宜及容易購買得到。這款錄音機是採用簡單的DC偏壓及DC抹音。因此，它能提供優良的錄音技術。

另一方面，這錄音機還有其它的好處；它擁有內藏AC電源及遙控裝置。還有，接連設備有DIN插掣。至於錄音帶計數器，亦是錄音時方便錄音帶定位的一個好帮手。

錄音機是附送有電路圖的。這本電路圖是有助於我們進行些微電路更改，以改進頻率響應。此等更改包括有除去C7（一個0·0033uF電容器是位於重插均衡器）。不過，這修改絕對不會影響錄音。

插入DIN插掣至錄音機是不會令揚聲器之電路斷開。要消除聲音，就要插入一個10Ω僞負荷耳塞至有關插孔。要特別留意的是：錄音機一定要遠離任何磁場，尤其是監視器之線路變壓器。筆者本人所擁有的錄音機，發覺有一難題存在。當錄音機處於重插及錄音狀態時，而按下暫停掣後，訊號幅度急劇性地下跌。筆者曾詳細檢查電路，但找不出毛病的因素，故只希望此僅爲個別錄音機的內裏小毛病，而不是線路的設計問題。

介面選擇

有三種介面可供選擇：TTL、局部RS232C及完整的RS232C。

在TTL的介面內，要預定沒有輸入及輸出緩冲的設備，同時需要一條短的電線（約0·5M）。如果採用TTL介面，錄音機的性能會被埋沒。但它能適合多種電腦使用。當特別的MC1488及MC1489IC購買不到時，RS232C介面配合晶體管工作是可作爲另一個選擇。這款介面，雖然頗爲實用，但它不符合RS232C的需要。無論如何，筆者雖十分成功地使用這介面，不過，採用RS232C接收器及傳送器IC是較爲合適。

當MC1488推動器及MC1489接收器於RS232C時，介面便能符合標準規格。長度達15M的電線亦可採用，當然，對地電容不能超過2·5nF限制。補充一點，MC1488及MC1489是Motorola公司的出品，如購買不到，可用Texas儀器公司的SN75188及SN75189代替。

電源供應

不同介面需要不同的電源供應。TTL介面爲+5V±5%，電流爲50MA。晶體管RS232C介面需要電源爲+10V及-10V，電流爲20MA。當RS232C配合特別IC時，電壓要在+8V至+14V及-8V至-14V之間。

適合的電源供應可在電路圖中見到。它是所有介面都合用的。在這個製作中，它亦被使用。

調校方法

要測試準確，最理想是使用視波器。如果沒有這個儀器，可用頻率計算器及模擬常用錶代替。但如擁有視波器的話，可依下列方法調校：

- 先接通元件，保證電源供給爲DC0
- 將R／W掣定爲（W）寫的一檔，然後連接頻率計算器至TP1及調校載波頻率至6·1KHz ±100Hz（利用預置之電位器，VR2值約6）。假如希望錄音帶能同時適合其它電腦使用，就要先讓元件有30分鐘的升溫時間，才進行上面提及的調校，同時，在這情況下，頻率誤差應降至±50Hz。如沒有頻率計算器，視波器的時基裝置可用來調校正確的載波周期，時基值約爲164Ms。
- 將DIN插掣入卡式機之接連處。R／W掣則撥至讀（R）的一檔，將續錄音幾分鐘後，重播錄音帶，而同時監視視波器顯示的串行資料輸出端情況。此外，還要沿順時鐘方向旋轉卡式機之音量控制直至隨機脈冲出現。然後，將音量控制況反時鐘方向轉動直至全部脈冲消失。最後，將制旋轉多數度，使獲得最佳的噪音邊界。
- 供給一個600Hz對稱之方形波至串行資料輸入端及將幅度調校至所選擇之介面電平。大部份之音響產生器皆可產生此訊號，但如讀者使用TTL介面，便要使用TTL訊號。（註：RS232C訊號電平約在-5V及-15V和+5V至 +15V之間）

• 將R／W掣撥至寫及錄音數分鐘。

• 將R／W掣再撥至寫的檔，重播錄音帶而同時監視串行資料輸出端於視波器的顯示。調校預置的電位器VR1至得到對稱波形。在這情況下訊號應該穩定，機訊號不穩定，可沿反時針方向旋轉音量控制少許。當實際應用時，音量控制之位置不是如此絕對，因爲幅度轉變達80%亦能容許。

沒有示波器的調校方法

• 將DIN插掣插入磁帶卡式機。R／W掣則撥至讀的一檔錄音數分鐘。然後重播錄音帶而同時利用一模擬伏特錶來監視串行輸出。將音量控制沿順時針方向旋轉直至能在電錶中見到隨機的波動。在這個過程後，再將音量控制沿反時針方向轉動至波動停止。最後，將制旋轉多數度，以得到最佳之噪音邊界。

• 將R／W制撥至讀的一檔。於重播卡式帶的同時，利用伏特錶監視TP2。然後調校預置之電位器VR1至讀數爲+2·2V。這2·2V是對稱方形波之平均電壓（假設峰值爲4·4V）。但要注要這調校未必能使錄音帶可用於其它之電腦。另外模擬伏特錶顯示的應爲平均值而不是峰值。

可是，伏特錶顯示的峰值會隨不同的IC改變。還有電錶所示的最低值約是0·2V，此最低值亦受不同IC的影响。故此，這電錶的固有內在誤差應計算一起，以便獲得準確的結果。

製作的要點

製作是需在一模型電路板上接線，當設計電路板時，應小心電路的佈置。放大器及單穩態（LM3900，74121）線路應當作單獨之件處理，而用獨立之電源。同時，在這個製作中，要使用到多個0·1uF去耦電容。但是在線路圖中，電容數目沒有明確列出，這由於是其數目決定於電路的佈置。要記着的原則是，每一個IC應有一電容接至其電源管脚。不過，這並不表示一個IC一定要有一電容，因爲兩個相近的IC可同用一電容。

通往單穩態計時零件的電路應該儘量簡短。另外連接15KΩ電阻的電源要直接接到隣近的去耦電容。至於置在電源供應的0·33uF電容，是用來防止不穩定狀態。還有，用於RS232C的電壓電路（+10V、-10V）要安排使避免與其它部份短路。

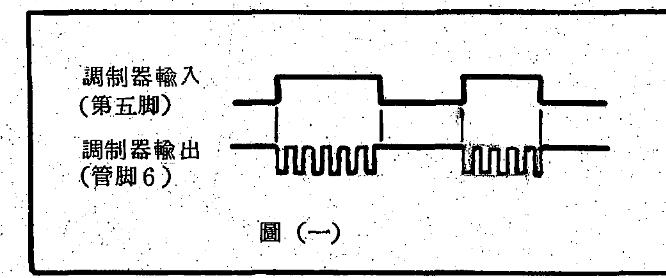
當調校完畢後，卡式介面已可使用。可嘗試接駁這1200波德RS232C介面及從串行埠輸出系列資料。然後，重捲錄音帶及再重播和將資料讀入。過程完畢後，可比較你原來所寫的，以作爲正確操作的測試。

零件表

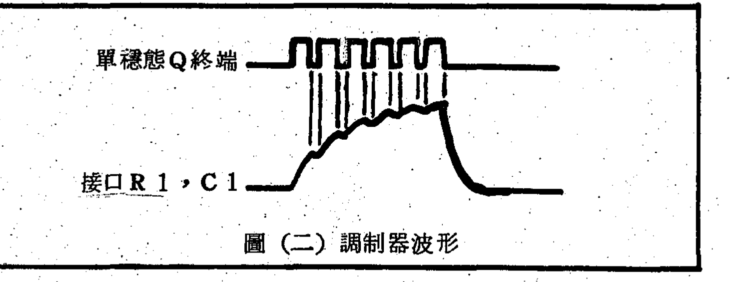
7.5－0－7.5V　IA變壓器
主要CABLE及插座
落地綫連接子一個
3脚DIN插座×2
SPDT選擇掣

半導體元件

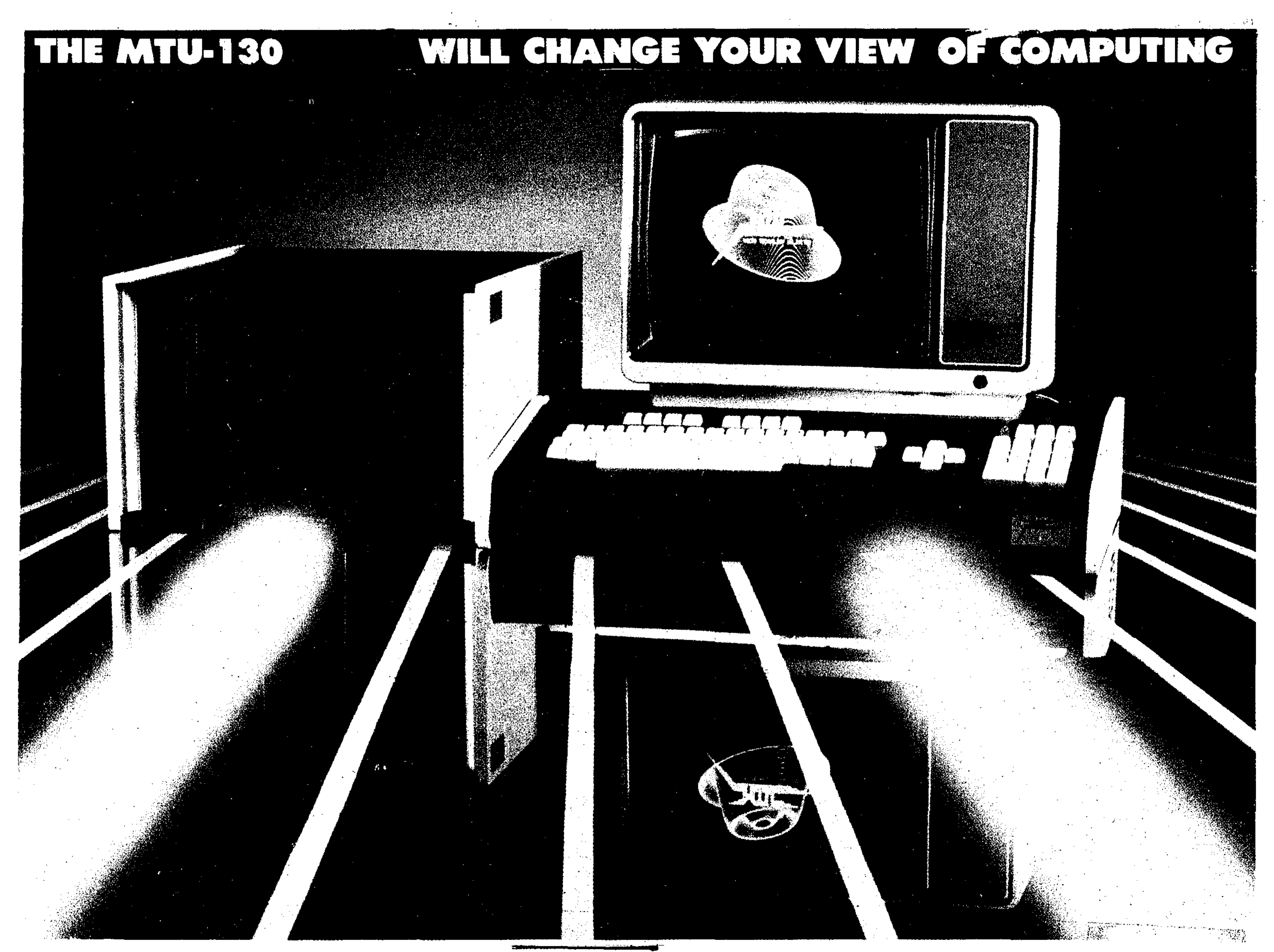
橋式整流或IN4002×4
IN914×3
7.5V　ZENER
BC557　PNP　晶體管
BC547　NPN　晶體管
7805+5V穩壓IC
LM3900　OP　AMP. IC
74121　　單穩態IC
74LS132　Z-INPUT　NAND　GATE
MC-1488　RS232C推動IC
MC-1489　RS323CIC



圖（一）



圖（二）調制器波形



新機介紹

資料分析功能

在過去的五年時間，MICRO TECHNOLOGY UNLIMITED正積極研究和設計6502的週邊機器。他們的累積經驗帶來了MTU－130的誕生

MTU－130具有256K RAM，地址是由內置的6502所控制。一個32K的程式從磁碟載入以至運算，只需要2·6秒，它的顯示圖像能力是480×256或240×256。MTU－130所採用的操作系統是COPOS——個類似UNIX的操作系統。

整個MTU－130系統可以分成三部份：首先是鍵盤部份，所有電子系統、擴展卡、都內置這個盒內，而其餘兩部份分別是兩個8吋磁碟機構成磁碟系統和CRT。將整個電腦系統分成三部份能幫助解決安放的困難。

至於應用語言方面，用者不需要擔心，MTU－130提供了多種語言給用戶選擇，其中的BASIC、PASCAL、FORTH、MACHINE LANGUAGE等。

首先讓我們看看BASIC，MTU－130所採用的是標準MICROSOFT BASIC。除了普通的指令外，還有新的指令控制ASCII檔案，光筆閱讀，功能鍵輸入等。最特別的設置就是LIBRARIES！

這組擴展命令可以與BASIC指令表（COMMAND TABLE）一起連接（LINK）起來。它們都是用機械語言所寫成，所以應用的速度很快。整個圖像（INTEGER GRAPHIC）僞圖像（VIRTUAL GRAPHIC）和CODOS介面程序庫（INTERFACE LIBRARIES）都是購買系統時附有的。怎樣編寫自己的資料庫也在說明書中詳細解述。

第二種MTU－130所支持的語言是6502支持的UCSDPASCAL。同樣，圖像指令也可PASCAL內設置，所以要得到高解像度的圖像是一件輕而易舉的事情。

除了PASCAL和BASIC外，MTU－130還可支持FORTH（Jack Brown形式的），利用CODOS介面和MTUEDITOR，使程式的大少失去了限制，令FORTH成爲一種甚具威力的語言。

一套系統的心藏除了微處理器外，便是操作系統。MTU－130所應用的是自行發展的CODOS。用戶需要更多空間的時候，磁碟的檔案便會巧妙地貯存在磁碟內，不致浪費空間。CODOS可以操作所有檔案，而且利用可改變的I／O通道（Channels）控制I／O裝置。所以無論你的程式是機械語言或一些高等語言，對於控制整個系統都非常方便。

該公司聲稱他們的CODOS是具備以下的特點：快速、具彈性和容易使用。CODOS的樣本是根據UNI—X的通道操作（Channel Operation）和在磁碟上貯存檔案所寫成的。因爲貯存檔案內容是以鄰接的形式出現，所以磁頭閱讀的速度十分快速。應用的操作方法可以是隨機（RANDOM）；順序（SEQ—UENTIAL）和其他提取方法。CODOS容許任何目錄的方式。無可疑問，CODOS是MTU—130更具吸引力。

如果你的興趣是在Programming，你必須注意MTU—130的MTUEDITOR，它的使用方法十分容易，它能夠容許用者兩方向翻頁（bidire—ctional scrolling）。你甚至不需要知道磁碟在應用與否。更能夠編輯1Meg byte的檔案！！不是比一般的文字處理軟件更具威力嗎？

聯絡地址：

MICRO TECHNOLOGY
UNLIMITED
2806 HILLSBOROUGH·
·PO BOX 12106
RALEIGH·NC 27605

價錢：

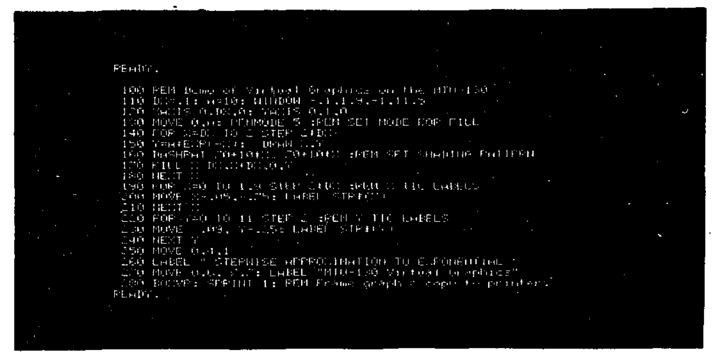
相信最多人關心的地方就是這部系統的價錢。筆者把MTU－130的產品價錢細列如下：

MTU—130價目表

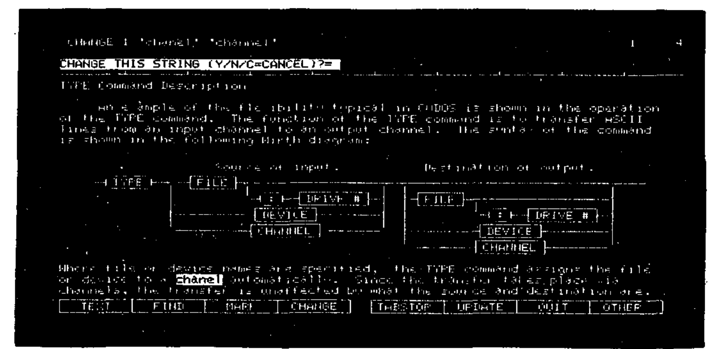
產品編號	說明	訂價 U·S$
MTU-130-2D	全套設備連兩部1M BYTE磁碟機及LIBRARY擴展 MTU-BASIC 1·0連GRAPHIC 12吋綠色顯像CRT及纖維光筆／兩個8吋雙面磁碟機	3599
MTU-130-1D	全套設備連乙部1M BYTE磁碟機	2999
MTU—130	電腦模板連標準磁碟操作系統，軟件及磁碟 控制器（不包括BASIC，CRT，磁碟機／機身）	1999
MTU-100-CRT	電腦模板連CRT（不包括磁碟控制器、BASIC或磁碟機／機身）	1699
K-1014-4	8吋磁碟機機身連電源供給器及接綫	1649
MTU-100	電腦模板（不包括磁碟控制器，CRT BASIC或磁碟機／機身）	1549
DATAMOVER	8MHz 68000 微處理電路板連256K BYTE RAM	999
2103-1	DISKEX	149
2101-1	MTU-BASIC連IGL、VGL、CIL	129
2104	ASSEMBLER/DIASSE LER	79
2102-1	MTU-130之音響儀器	49
K-1002-10	MTU-130圖像軟件	49
2200-1M	MTU-130（776頁）手冊	
	美國收書付款	40
	美國預付	35
	加拿大預付	45
	美加以外地區預付	75
	NEC8023A印字機連接綫及磁碟機	645

MTU－130
專用穩壓器

要保護用者所編寫的程式碼，MTU－130附設了一所工廠，它們都附一個唯一的串連號碼，只需要知道這個號碼是怎樣寫在程式內。每一個用者當採用他的軟件時，必須供給他的串連號碼。如果你是一個機械語言的程式設計員，MTUASSEMBLER必定成爲你的好帮手。一個218K BYTE的源檔案（SOURCE FILE）和800個代符（SYMBOLS）滙編成代碼，一個代符表，一個排列了（SORTED）的相互參考的地圖和列出檔案，貯存在磁碟裏，只需要4分鐘。除了這個滙編器外，還有一個選擇性的2716／2732 EPROM編寫器供給用家。

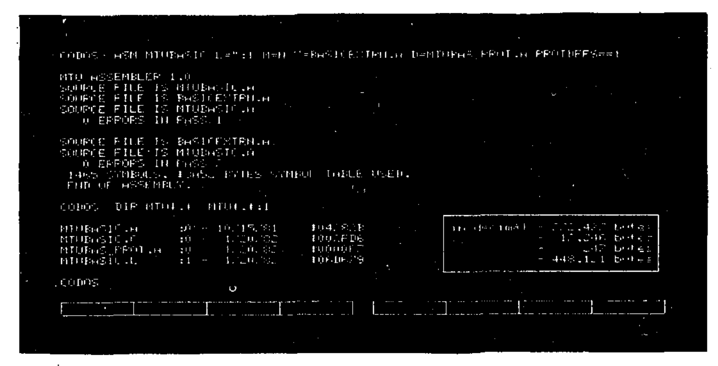


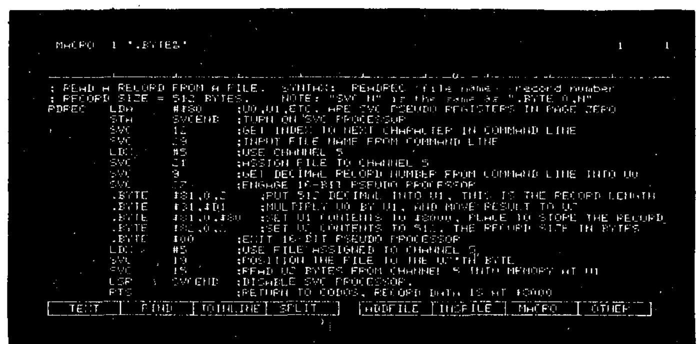
BASIC 語言



電腦系統組合

你可能希望你的電腦與其他的電腦溝通，互相交通文字和數據檔案。MTU－130同樣地能夠勝任。一個名爲DISKEX的軟件容許用者讀寫以IBM，CP／M，FLEX格式8吋磁碟內的ASCII檔案。當然，卡式埠容許從音響錄音機中轉換更多系統。還有，一個仿效終端機的程式（標準附設的），使MTU－130的操作成爲另一部電腦的終端部份。利用這個軟件，你可以利用CRT／鍵盤主機，或磁碟上的檔案，來傳送反接收訊息，或者是兩種合兼而得到一個完整的磁碟傳送。





MTU-130的擴展可能

原裝系統所附帶的標準磁碟控制器可以處理4個雙面的磁碟機（貯存高達4M BYTE）最大的好處；還是其餘的地址空間。意思是：將來你將會擁有更大的擴展能力。擴展的硬件包括：DATAMOVER的68000卡；4000×4000位置的VECTOK CRT，固定或可替換的硬碟系統，爲產生職業水準音樂的12 BIT D／A系統，高解像度彩色等。這些東西都能夠像內置的功能一般很具彈性地應用，因爲有足夠的地址空間。要機械性的擴展，內置的卡檔案具備了三個沒有用途的位置，可以容納三塊11吋闊的卡。如果你希望自己製造擴展器具，一塊選擇性的原板使裝造起來十分容易。

68000卡—外圍接合

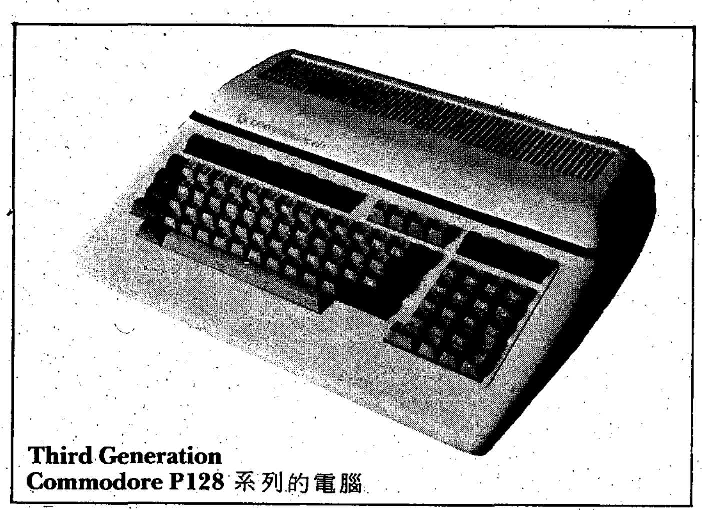
DATAMOVER 的68000使MTU－130更具威力。

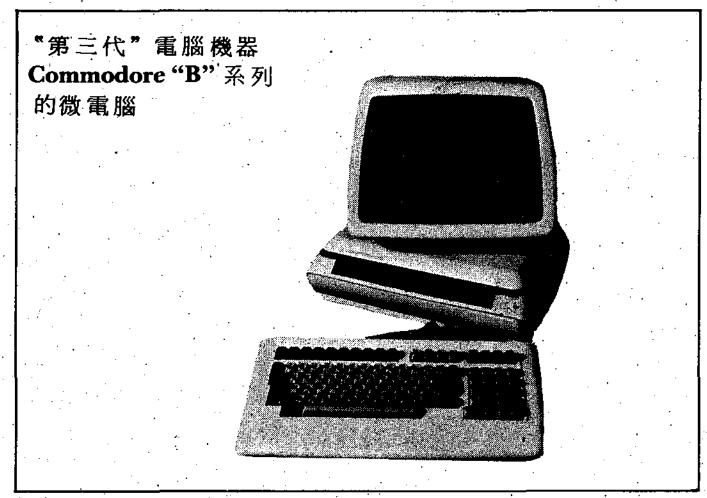
當你把這塊卡插置在MTU－130後，你的電腦便會全面性升級，自然它是和完先的系統完全相融的。你的6502現在可以得到8MHz，16BIT，68000微處理器威力。DATAMOVER的68000微處理器利用6502作爲前後（FRONT－END）I／O處理器，不但使運算能力增加，也使記憶量提高。這樣用者對於加上68000後，MTU－130不會產生陌生的感覺。CODOS，圖像顯示、光筆、鍵盤和其他I／O工作像以前一般，不同的是現在的威力比先前更大。

除了68000，DATAMOVER還包括了256K RAM，它能加上額外記憶擴展至1MEGABYTE。6502埠則具有16個數據BITS。這兩個處理器同樣應用共通的記憶。68000具有19BIT的地址線可以直接提取記憶。■

COMMODORE

第三代電腦


Third Generation
Commodore P128 系列的電腦


"第三代"電腦機器
Commodore "B" 系列
的微電腦

第三代電腦

COMMODORE 最近推出一系列第三代電腦生產計劃，如果照原定計劃，到今年年底，將會推出超過五款新三代電腦，COMMODORE今次推出不是一款，而是五款之多，照其公司所透露，是希望兼顧上、中、低價市場，在原來基礎上將產品升級，改良或減價等。

下價的有"MAX"，其美國訂價只不過是美金179·95，折合約千壹元左右，但直至目前爲止，其特性均未得悉。

在VIC－20改良的有COMMODORE 64，基本大變動是VIC－20加64K RAM。

一千元美金有"P"系列，有128K記憶，隱約見到未來MICRO的趨向。

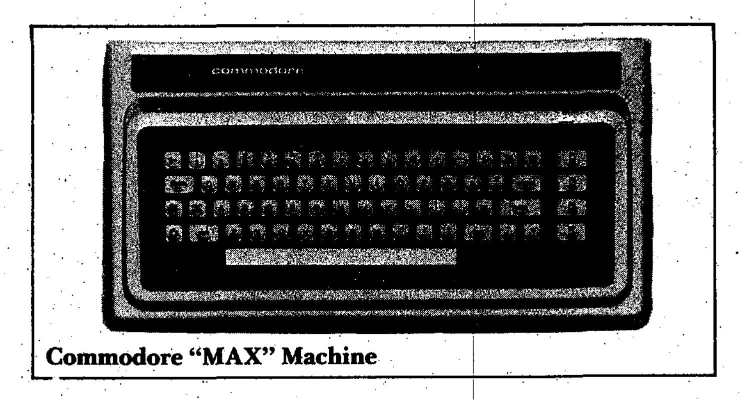
而放在辦公室的B系列，連同CP／M，雙磁碟，似乎想與IBM一較高下，但售價只爲美金三千元左右。

在目前議論紛紛的時代，一部基本電腦，要加上DISK，及其它附件，才可以稱得上用於辦公室，或其它較爲嚴肅的用途，爭論的目標是與其一件一件分開來購買，不如整件的購買可能更爲化算，但因各人的意見不同，很難將意見統一，COMMODORE便索性出多款機以迎合市場了。

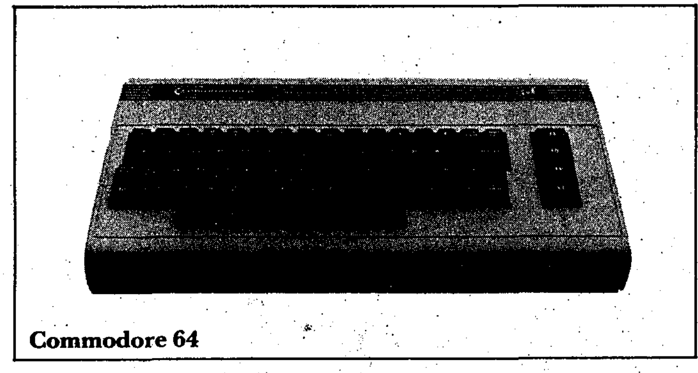
現在生產MICRO已經好像生產電風扇了，那一級有什麼設備已定型：低價的是怎麼樣，高價的又要有什麼附件。沒有人再買一套兩三萬連什麼軟件，或將操作系統另創一格的機種，OS與及記憶容量已分別定型爲一兩種。一部機如果沒有128K的休想在BUSINESS市場，而千元左右的玩家便必需忍受一個細小的鍵盤。

COMMODORE的多方面發展，今市場很震驚，但據該公司高層人員透露，要達到一個目標就是做一部全面性軟件匹配的MICRO，無論什麼軟件，APPLE、ATARI、TRS－80等均可在機上運行！其實這一點是不難做到，因爲現在記憶容量大，如果用16或32BIT的CPU，根本就無需什麼DOS（磁碟操作系統），通通都放在記憶中，由電腦的硬件去分配運行程式，如此這般，可以玩一輪CP／M，寫一輪PASCAL，做到半路轉寫BASIC，故此將所有APPLE，ATARI、PET等程式都可以搬到同一部機上運行的那一天相信不遠了。

圖右 價錢低於２００美元
的微型電腦



Commodore "MAX" Machine


Commodore 64

左圖是ＶＩＣ　２０的最新改良型

系列的特點

	BX256	B128	"P" Series	64	MAX
監視器	80X25 綠色	80X25	40X25 可外接彩色	40X25	40X25
雙磁碟 可配卡式	有	有	可選	可選	
基本 RAM	256K	128K	128K	64K	未悉
彩色數目	N/A	N/A	16	16	未悉
圖象分解度	N/A	N/A	320X200	320X200	未悉
可否相容ＣＰ／Ｍ	可以	可選	可選	可選	N/A
IEEE-488介面	可以	可以	可以	一款與ＢＡＳＩＣ４·０大致相容的ＢＡＳＩＣ	N/A
RS-232	有	有	有	未悉	N/A
標準語言	BASIC 4.0	BASIC 4.0	BASIC 4.0	A BASIC	用ＢＡＳＩＣ卡式匣
(美金·美國本土) 計劃零售價：	$2995	$1695	$995	$595	$179.95
推出日期：	秋季	秋季	秋季	７／８月	

VIC-20



相信很多讀者已知道ＶＩＣ－２０的存在，但到底其性能是怎樣的？筆者最近曾詳細試用一部價錢低廉的ＣＯＭＭＯＤＯＲＥ ＶＩＣ－２０家庭式電腦，發覺ＶＩＣ－２０是一部有多方面用途的機器，以價錢論之，更是物超所值。

ＶＩＣ－２０外表是白色膠殼，非常簡潔，體積是闊４００ＭＭ，深２００ＭＭ及高６５ＭＭ（以機背後計算，因爲機背方面較前面爲高）。面板包括６２個標準按鍵及四個可程序式按鍵。電源接線，電源開關制及遊戲盒帶插入埠均位於右面側；串聯埠（ＳＥＲＩＡＬ ＰＯＲＴ），錄音機連接座，用戶埠（ＵＳＥＲ ＰＯＲＴ），影像機接座及擴展埠（ＥＸＰＡＮＳＩＯＮ ＰＯＲＴ）均位於背後，而擴展埠主要是插用ＣＯＭＭＯＤＯＲＥ的盒帶程序（例如遊戲盒帶，包括有ＰＡＣＭＡＮ及ＲＡＭ之擴展盒帶）。

ＶＩＣ－２０包裝得非常安全，用發泡膠盒將整部電膠包裹着，電源供應、影像調節器、映像傳輸線及一本厚達１６４頁的ＶＩＣ－２０應用手冊均被泡沫膠分隔開，以免因運輸上之碰撞而受損。ＶＩＣ－２０之安裝法則詳細列於另一本小册。

電源供應部份是利用變壓器將２４０Ｖ交流電降壓成１１Ｖ之交流電，由於變壓器本身並沒有開關掣，所以ＣＯＭＭＯＤＯＲＥ廠忠告用戶如不是用電腦，必須將變壓器之電源線拔掉。ＶＩＣ－２０之開關掣細少，是否開着，全靠面板上一盞非常小之紅色ＬＥＤ燈所顯示。

映像訊號經由電腦背後之五針插座輸出，金屬包裝之電視訊號調節器插於這五針插座上，再經由一條訊號傳輸線接駁至電視機之天線插座，調節器上則有一個選擇掣，可選擇ＸＨＦ訊號之０或１頻度。

ＶＩＣ－２０之軟件盒帶是插入電腦右側之擴展埠，錄音機埠則是接駁至錄音機，但沒有供給錄音機馬達開關之用０另外，ＶＩＣ－２０通過串聯埠（ＳＥＲＩＡＬ ＰＯＲＴ）接駁至印字機或磁碟機，但應用手册上沒有詳細解釋串聯埠之應用（這可能是屬於特別配件所置）。

從應用手册之輸出圖，顯示出ＶＩＣ－２０可共接三支歡樂棒（ＪＯＹＳＴＩＣＫ），一枝感光筆，一個錄音機馬達控制配件，一個串聯配件及一個復原（ＲＥＳＥＴ）設備。這個接駁器（ＣＯＮＮＥＣＴＯＲ）一半是屬於ＶＩＡ（多用途接面轉換器），可用作雙向平衡埠，作雙向握手式訊號傳輸（ＨＡＮＤＳＨＡＫＩＮＧ ＳＩＧＨＡＬＳ），接收與輸出訊號同時進行。

應用手册只列出ＰＩＮ－ＯＵＴ（輸出接駁）的資料，並沒有詳細解釋如何使用，但我們却可從〝ＶＩＣ ＰＲＯＧＲＡＭＭＥＲＳ ＲＥＦＥＲＥＮＣＥ ＧＵＩＤＥ〞中獲得這些資料，這本書是須要另外購買的。ＶＩＣ印字機（編號ＧＰ８０，屬於點陣印字機）及ＶＩＣ錄音機的資料，則詳列於應用手册裏。

ＶＩＣ－２０圖案

最值得注意一點是ＶＩＣ－２０之顯示（ＤＩＳＰＬＡＹ）質素非常差，筆者曾利用調節器經由彩色電視機作輸出螢幕（這是一般用家所採用的方法），當選用１號頻度時，電視之畫面仍算可接受，但沒有彩色；倘若選用０號頻度時，畫面雖然有彩色，但色彩不斷閃動，令眼睛難以接受，如經過細心調節電視機之調頻器，效果會略佳。

據稱ＶＩＣ－２０擁有一本技術手册，教導如何於ＶＩＣ－２０內作適當之調較，致使影像方面較爲理想；就算是選用ＲＦ調節器（ＭＯＤＵＬＡＴＯＲ）亦而然。

對ＶＩＣ－２０而言，利用調節器經由電視機顯像，黑色字體的外圍被閃動之黃色包圍着，令眼睛非常辛苦；除此之外，當作一些顏色組合時，亦有同樣情形發生。上

述缺點成因主要有兩點，其一爲傳統性電視機之接收器接收範圍受到限制；其二則爲VIC－20之RF調節器有一定之限制所致。

筆者曾利用彩色監視器（COLOR MONITOR）作爲VIC－20之顯視用，無論影像及聲響均十分理想。最近有一款35CM之彩色電視接收系統，適合VIC－20使用，價錢爲379美元。

如果是用彩色監視器的話，VIC－20之彩色及圖案可算相當理想，但仍存有其他缺點。

VIC－20字源顯視於屏幕上共爲23行，每行22個字，字源體積非常大，而整個螢幕表面，VIC－20佔用約3分之2，其餘3分之1剩餘部份則爲顏色所蓋。

英國COMPUTING MAGAZINES報導，每行40字之"VIC－40"將於短期內推出市場（COMMODORE於英國的市場是最大的），這將彌補VIC－20每行22字之缺點。雖然VIC－20編寫程序時可超過一行，但由於字體龐大，致令閱讀方面非常困難。

如包括大寫及小寫，VIC－20共提供66個圖案字源。按動SHIFT鏈時即令VIC－20進入圖案狀態（圖案形狀已印於每個按鍵前面之右角）；當按動另一標着"COMMODORE LOGO"之鏈時，VIC－20即進入另一組圖案字源狀態（圖案形狀已印於每個按鍵前面之左角）；兩個一起按動時即進入UPPER/LOWER CASE狀態，直至兩個鍵再被按動爲止，才回復原狀。

當LOWER CASE狀態時，只能選用字鍵左角之圖案字源；同時，於此小寫狀態下，影像亦轉至小寫，但却可利用SHIFT鎖掣而打出每個字或行之UPPER CASE（大寫字體）。

VIC－20鍵盤與標準形式打字機之鍵盤排列形式相同，所以非常容易使用。這點似乎COMMODORE公司於生產以計算機鍵盤形式排列之PET時得到教訓。

浮標控制鍵是自動重複的，所以移動浮標以編輯（EDITING）程序非常容易。除了浮標之移動外，程序內之字源亦可於螢幕上自由移動以作應用；當利用這種方法時，必需按入"PRINT"，浮標控制代碼則以不同之逆轉圖案字源形式顯示於螢幕上，而空位鍵亦同時變成有自動重複之功能。

當VIC－20一經開着式復原（RESET）後，電視螢幕即變成白色而四周則爲青色，字體則爲深藍色。鍵盤上最高一行按鍵由1至8是專責控制字源顏色的，當按下CONTROL KEY後再連同其中一個鍵按下將能改變字源的顏色，顏色包括有黑色、白色、紅色、青色、紫色、綠色、藍色、黃色等共八種顏色。

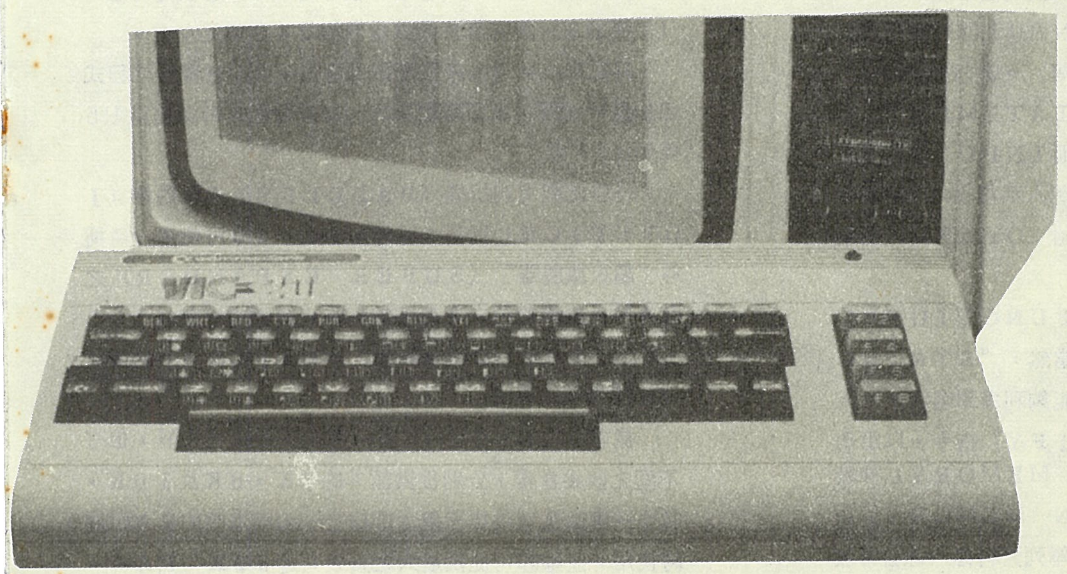
浮標控制鍵一樣，顏色譯碼能於編寫程序時一起編入，利用"PRINT"而改變字源於螢幕上之顏色。如將CONTROL KEY與RVS ON一起應用的話，畫面底色與字源顏色將即互相對調換，直至到按入RVS OFF才能回復原狀，這項功能亦能同時被編入程序內。

基本上，VIC－20內有三個地域的顏色是互相關連的，我們可從螢幕中見到，當字源顯現出來的時候，字源的顏色並不會與屏幕中非現字部份（四周之顏色）之顏色相同。非現字部份共有8種顏色可供選擇，而顯字部份則有16種顏色。

現字部份及非現字部份之顏色共有255種顏色組合，決定後將該顏色組合之譯碼編入36879之記憶位置；試舉例做出一個兩者之間的組合：POKE36879，122，將會將現字部份編成淺黃色，四周非現字部份則編成黃色；POKE36879，255則將現字部份編成黃色，四周非現字部份却變成淺黃色。顏色組合之詳細資料可於應用手册37頁找到。

將圖案及浮標控制字源編入程序內，可令整個程式更有生氣及美妙，產生一系列之閃光及畫面改變顏色，最適合用於編寫遊戲。

POKE之另一個用途是於圖案功能時，直接將字源密碼編入螢幕記憶內。由於應用到一個特殊之影像接面晶片。（VIDEO INTERFACE CHIP），所以VIC－20擁有一個非常特別之記憶結構，這個VIC（VIDEO INTERFACE CHIP）是COMMODORE特有之晶片，是專責控制影像之顯現及音響效果。



螢幕上共506個位置，都必須用兩個記憶數字來代表。螢幕記憶位址是於7680（10進制）開始；每一個記憶位置所記憶的是於螢幕某一指定位置上所必須出現的字源或符號。另外一個記憶區，共有506個拜特之記憶單位，是由位址編號38400（10進制）開始，用以決定螢幕上某指定位置之字源顏色。

聲響效果

VIC－20之聲響效果是特別的，提供四個聲音（VOICE），三種聲調及一個白噪音（WHITE NOISE）；能夠獨立地產生單聲調或聲調與噪音之混合，共超過三個範圍之第八音。

四個聲音來源之音量控制，是經過一個變數POKED入記憶位址36878內。音量範圍是由0無聲至15最大聲，當然亦可以經由電視機之音量控制旋鈕直接控制。

音響效果千變萬化，且於應用上非常容易控制。VIC之應用手冊內提供一系列之程序以供用者試奏出一些美妙之音樂，另一些例子則是教用者直接按鍵而發出音調。於另外一本小冊子上，共有20個例子程序，給用者試制一些聲響，如音樂比率，鳥鳴、UFO之降落、投擲炸彈聲音、炸彈爆炸聲等等。

VIC－20 BASIC

VIC－20所用之BASIC語言，與其它的COMMODORE機器相去不遠，一個有用之陳述是"CMD"，是將螢幕之輸出再次帶往其它外接設備，其用途如下：

OPEN－1，4表示打開4號設備，這設備是一個印字機；

CMD1－表示將所有輸出引領至印字機；

LIST WILL－將程序內容詳列於印字機上，並非螢幕上；

CLOSE1 WILL－將輸出引領回螢幕上；

其它設備亦可同樣利用OPEN而達到開關應用之目的，（其中包括磁碟機，錄音機等等）。當外接設備一旦開始（OPEN）後，陳述句（STATEMENT）GET#將準許每次輸入一個字源；而INPUT#則準許由特定之設備內輸入不同數量的字源入到程序內。PRINT#是用作將變數送往一個已開始（OPEN）之檔案（FILE）或其它設備去。

預定功能（DEFINED FUNCTION）亦同樣可以於VIC－20上應用，當然，"STRING"功能亦不例外。錄音機之操作是能夠利用到名字（NAME）作輸入；更包括有"VERIFY"指令，以用作核對由副記憶（SECONARY MEMORY）抄錄至電腦的主記憶上之程式（MAIN MEMORY）是否有差別。PRINT#是用作將陣列資料存入錄音帶裏。

另一項有趣東西是VIC－20內之時鐘，VIC－20保留了一個"TI$"以用作眞正的時間計時，準確至以秒形式顯示出來，但當VIC－20開機時即將時間自動清洗，而用者却可隨意將時間較至正確讀數去，之後，VIC－20又再能從新以時、分、秒將準確時間顯示於螢幕上。這個REAL TIME CLOCK（時鐘），同時是可以用作時間延遲（TIME DELAY）及遊戲之計時用。

整數（INTEGER）及實數（REAL VARIABLE）（均是最高去至9位），多重立體陣列均能於VIC－20上使用，於表（一）裏我們可以見到VICBASIC被詳細列出。

VIC BASIC之編輯功能（EDITING FUNCTION）非常好用，當一整個程式列出後，那些浮標控制鍵（2個鍵及移動鍵）能夠控制浮標至所要更改的一行；而只要將新的一句印於舊的一句上，再按RETURN即可同時將舊的一句抹去而記入新寫入的一句。

機械語言用USR及SYS所支持，USR（X）是控制機械語言程式至記憶位址0001及0002，X之數值，是當作參數通過至程式裏。一個10進位數值或變值（VARABLE）跟隨着SYS（範圍由0至65535）是將控制交至機械語言程式或所指定之的記憶位址去，但却阻止參數通過至ROUTINE，如用者想更清楚這些命令（COMMANDS）的詳情，可參看引用手冊。

程式盒帶

VIC－20擁有一系列之盒帶程式（需另外購買），可隨時於電腦主機後面插入。筆者曾試玩太空船降落火星遊戲，初時是頗具刺激，但很快即覺呆板，主要是每次遊戲之經歷均一樣，很快便能克服該遊戲。其它遊戲包括"SUPER ALIEN"（超級異形），"DRAW POKER"，"MID NIGHT DRIVE"（午夜飛車）等。每盒遊戲盒帶約爲40美元左右。

其它程式更可以錄音帶形式購買，其中教育性之程式包括數學練習，家庭財政管理。錄音帶程式每盒約爲15美元。

一些遊戲可利用撓（PADDLES）式歡樂棒（JOYSTICK）玩耍。另外VIC－20可使用一種稱爲"超級擴展器"（SUPER EXPANDER）之RAM，提供額外之3K記憶拜特高解像力畫圖（1024×1024點）及範圍廣闊之BASIC命令，這個盒帶售價爲69美元。

標準形式的VIC－20只可提供5KRAM，但有不同之擴展盒帶，分別提供額外的3K，8K及16K，其中一優點是可將該三個盒帶結合一起與本機的RAM同時使用，這樣即可提供最大之擴展共達32K RAM。

PRICES

有一點令顧客却步的是，所有VIC－20之配件，只有COMMODORE廠方之產品供購買，這由於VIC－20之錄音機，磁碟機及印字機等所沿用之介面（INTERFACE）IEEE，較少爲其他廠家所生產，所以顧客並無他選，必須購買COMMODORE之配件。

COMMODORE之錄音機，售價99美元，比其它貨廉物美產品貴很多。COMMODORE磁碟機，連同控制線路（CONTROLLER），售價爲699美元，採用14CM磁碟，記憶量爲170K拜特。

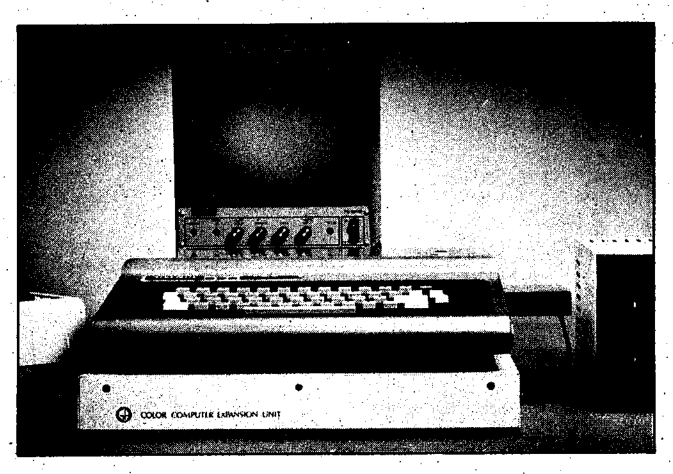
VIC－20主機本身，DICK SMITIH ELECTRONICS PTY LTD·之售價爲399美元。VIC－20對初學者而言是一部適合的電腦，雖則一些廣告標榜VIC－20可作商業用途，但筆者對這點存有很大的疑心。

如VIC－20與RADIO SHACKS之TRS－80彩色電腦比較，兩者之性能與價錢相若，要下結論頗爲困難。如撇開顏色監視器，VIC－20本身則較TRS－80略爲便宜。VIC能產生更多之顏色，但標準形式欠缺高解像力圖案功能。另一點是，雖然VIC控制容易，但其聲嚮效果非常成功，這一點TRS－80望塵莫及。

用6809微處理機與VIC的6502微處理機比較，就相等於以計算機方式之按鍵排列對標準形打字機按鍵排列的運用，結論由讀者自行決定。無論如何，VIC－20是值得考慮的。

一新產品一

電腦彩色的擴展裝置



COLOR COMPUTER EXPANSION UNIT

RAM 記憶容量到64K，提供RAM用戶由零到61·5K記憶容量。

它包括一部Z80A的微處理器，而這部Z80A微處理器跟COLOR COMPUTER的6809微處理器一樣，可運行CP／M，FLEX和OS－9程序。

用戶能夠經由COLOR COMPUTER的內部顯示和一80×25的字母數字顯示來選擇軟件。而這軟件可作反向顯示，雙密度，閃亮／間隔，倒置和保護字符。

一雙密度的5¼吋磁碟控制器，能夠支持四個推進器具有800K的記憶系統，而能貯存一3·2百萬的貯存容量。

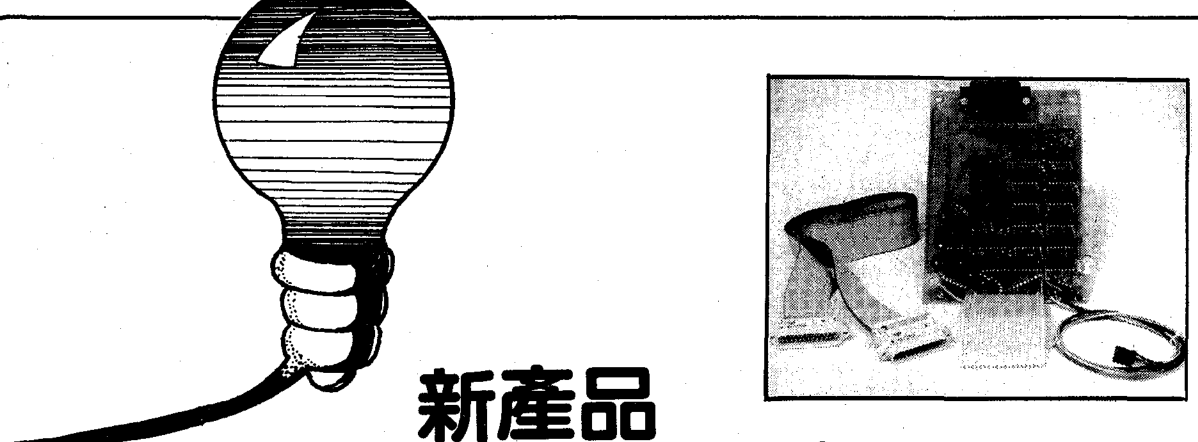
附加特徵包括一COLOR COMPUTER可兼容RS－232C串行埠，一可選擇的IEEE－488／1980控制器，而與數字的週邊和實驗室儀器，互相連接，一可選擇的光筆和兩個擴張總線，用來作外部線路的附加。其一內置的聲音推進器和揚聲器，而容許COLOR COMPUTER的聲音輸出。

這單元加插入不用改變和容易刪去，COLOR COMPUTER盒式磁帶埠可容許獨立使用。

缺少IEEE控制器的，售價爲1585美元，而具備有IEEE控制器的，售價則爲1750美元。

這COLOR COMPUTER EXPANSION UNIT由GEORGE ASSOCIATES公司出產，地址爲：

P·O·BOX 960，BERKELEY，CA 94701，(415) 843—3587·



新產品

ucts...new products...new

APPLE 232C I/O卡

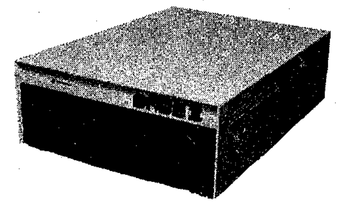
PDA 232 C I/O卡是一多方面的輸出輸入串列卡，而適合於APPLE II 應用，它能提供三個操作類型：輸出輸入、終端裝置和選擇程式。

這個介面加上不同的 RS－232C裝置，即如；印字機、終端機和解調器，這卡擴展到你的 APPLE 是相當適合的。售價爲加幣179元。

聯絡地址：

INTERPLANETARY COMPUTER SYSTEMS LTD.，

950 DENISON ST.，UNIT 17，MARKHAM，ONTARIO L3R 3K5.



68000的SAGEII系統

最近SAGE COMPUTER TECHNOLOGY 公司介紹的SAGE II，是以68000爲基本單板電腦作藍本。

SAGE II 能夠應用PASCAL、BASIC、FORTRAN和一些滙編語言。

這部SAGE II 的記憶，可由128K到512KBYTE的RAM和在兩個5¼吋的軟碟磁碟機內，而可增加至1.3兆波特。

包括兩個RS－232C的串行埠、一可與CENTRONICS 相容的平行埠和一部IEEE—488的介面。

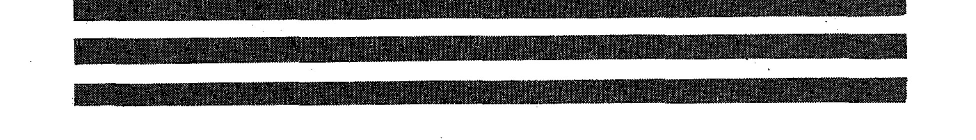
SAGE II 的售價約爲3600到6000美元，那便要視乎它的磁碟和隨機可存記憶（RAM）裝置而定。一部可供選擇的中斷式推動的UCSD 操作系統，售價約爲400美元。一部附加128K BYTE的隨機可存記憶售價亦爲400美元。

如欲知詳細的資料，可致函到：

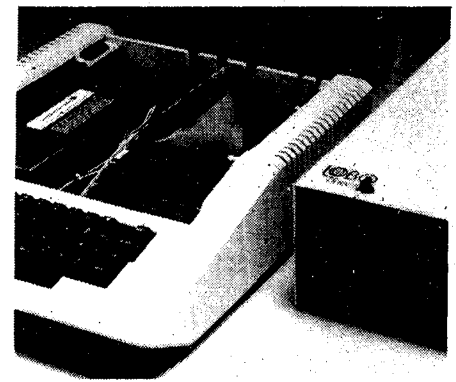
SAGE COMPUTER TECHNOLOGY，

SUITE 14，195 NORTH EDISON WAY，RENO，

NV 89502，(702) 322-6868.



CP/M FOR APPLE



LOBO DRIVES推出一些硬件和軟件，它是作爲 APPLE Ⅱ使用的CP／M。標準的迷你軟碟，LOBO 8吋的單或複邊雙密度的軟碟系統和LOBO 5¼吋或8吋的硬磁碟系統。

這部LOBO CP／M系統特徵，包括：不會減小用戶的RAM面積、能夠提供LOBO 5202和8202的軟碟系統，支持 LOBO 950和1850的硬碟系統，提供APPLE迷你軟碟的標準和能夠在同一時間內在每一行的每一線上。

如欲查詢更詳細資料，可去函：

LOBO DRIVES，354 S．FAIRVIEW AVE．，GOLETA，CA 93017．(800) 344－6103或(800) 344－6104 FROM CALIFORNIA；OUTSIDE OF CALIFORNIA (800) 235－1245。

可變字型的軟件

A friend is one To whom one may pour Out all the contents Of ones heart, Chaff and grain together Knowing that the Gentlest of hands Will take and sift it, Keep what is worth keeping And with a breath of kindness Blow the rest away. - ARABIAN PROVERB	A friend is one . To whom one may pour Out all the contents Of ones heart, Chaff and grain together Knowing that the Gentlest of hands Will take and sift it, Keep what is worth keeping And with a breath of kindness Blow the rest away. - ARABIAN PROVERB

CHARGEN 產生三個不同的尺度的SCRIPT，OLDE ENGLISH，和ROMAN BOLD 字符，在ANADEX DP－9000 串行印字機。這實用程序能夠用在存取文字處理軟件，用來適合TRS－80型電腦。

在操作過程中，你能夠用文字處理程序去產生原文文件，和存貯在磁碟內，然後CALL CHARGEN。

利用選擇適當的印刷參數，印出所要求的字符字型。

這程序是利用機械語言編寫和最少需要32K RAM及一磁碟推進。

如欲知悉更詳細的資料，可函：

JOE KING，7342 CAPISTRANO AVE．，CANOGA PARK，CA 91307．



tv monitor

BYTM 系統公司宣佈推出名爲"CLEAR STAND"的電視／監察器，適合用在ATARI 400／800 和APPLE Ⅱ上。

它是用高冲擊力磨光的丙稀酸製成的。屏幕最大可增至15吋。

售價爲59．95美元。

如欲知詳細資料，可函：

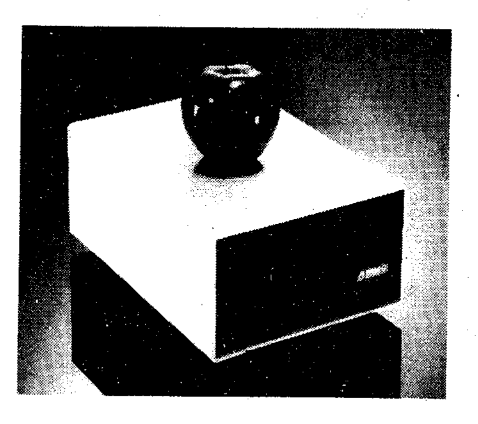
BYTM SYSTEM，INC，389 FIFTH AVE．，(SUITE 400)，NEW YORK，NY 10016．

❀❀

apple diskdrive

A．M．ELECTRONICS INC．宣佈一部5¼吋的磁碟機供應給 APPLE 電腦系統。

40軌的售價爲美金395元；而80軌售價約爲495美元。



如欲知道更詳細的資料，可函：

A．M．ELECTRONICS INC．，3446 WASHTENAW AVE．，ANN ARBOR，MI 48104。(313) 973－2312．

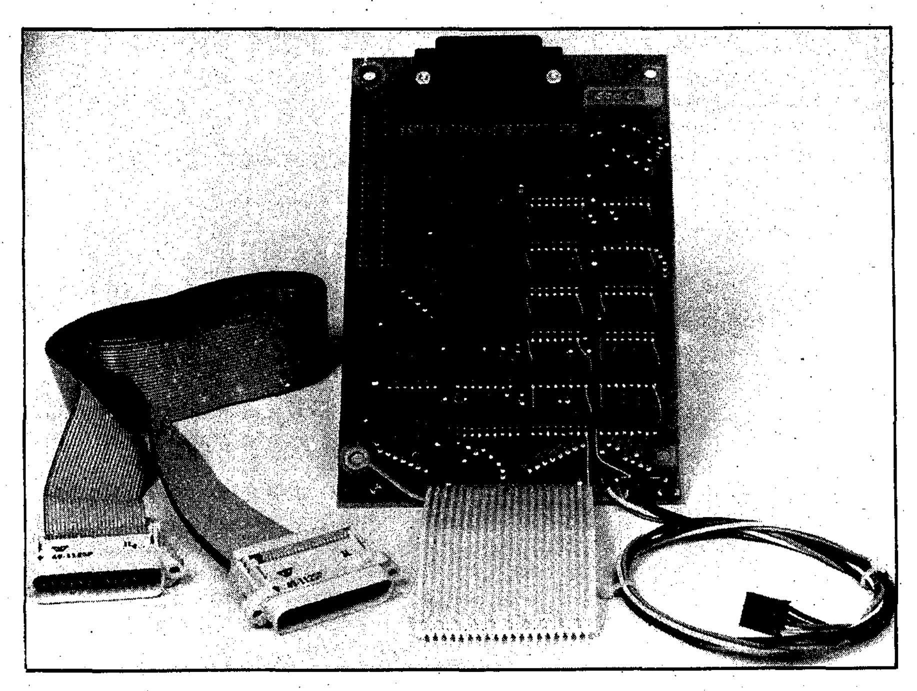
RS-232C 卡

VR－RS－232C是一插入式的串行介面，它可適用於TRS－80 MODEL Ⅲ。具有50－19200鮑速（BAUD RATE），程式可以稍出（PIN OUT）消除至援綫纜（即MODEM OUTPUT STANDARD），設有反向極性保護、UART和給予用家特別應用的原型區段（PROTYPING AREA）。除此之外，本身還有20毫安迴路電流作繼電器或亮燈之用。

查詢資料可與下列公司聯絡：

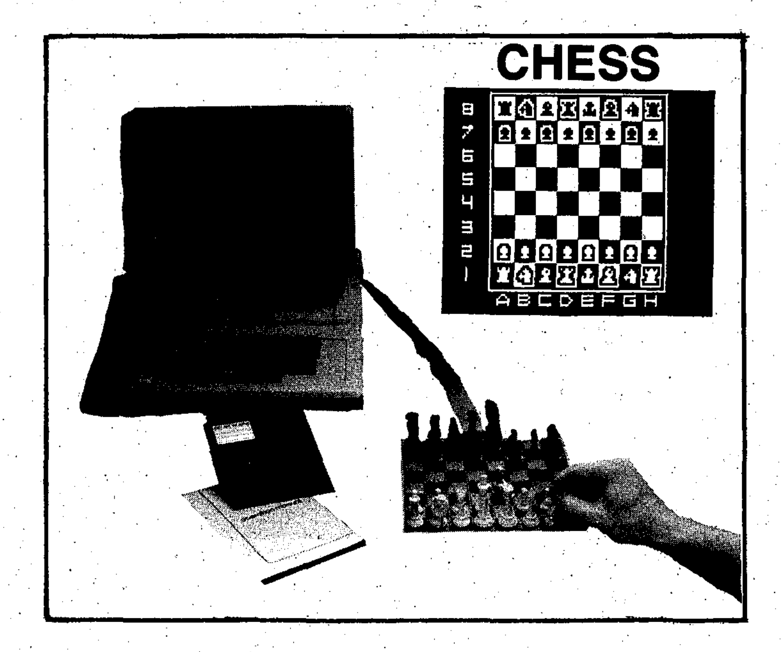
VR DATA CORP. ，777 HENDERSON BLRD，N－6，FLOCROFT INDUTRIAL PARK。

PA 19032. (215) 461－5300



新產品

國際棋周邊附件



國際棋週邊附件

MATE，是APPLEⅡ的硬件和軟件的週邊附件。它包括一個強大的下棋程序，它是附合製造商的要求而在微型電腦入發展的。

這部週邊附件是特別爲32K－BYTE APPLE，TRS－80和PET電腦而設計。

MATE供應一部磁感應的棋板、磁性的棋子、插入卡式面板和一些接纜。

這部國際棋包括有很多的下棋特徵，包括有：對付〝敵人〞的思考能力、九級難度下棋階段、一個公開的貯藏6000個行棋步法的記憶系統、最完美的隨機行法選擇、行棋的意見、反白的棋盤和印出完整的遊戲步法。在介面軟件上，可以印出完全的文件編制，這是提供一些用戶製造其本身的國際棋程序。

這部MATE的製造商是APPLIED CONCEPTS公司。售價約爲269. 95美元，包括卡式和磁盆式的遊戲程序和人工部份。

如欲購買或查詢更詳細的資料，可去函：

PMK ASSOCIATES，
POB. 598
EAST BRUNSWICK，
NJ 08816，
(201) 246－7680。

新產品

"Internal Memory"的擴展系統

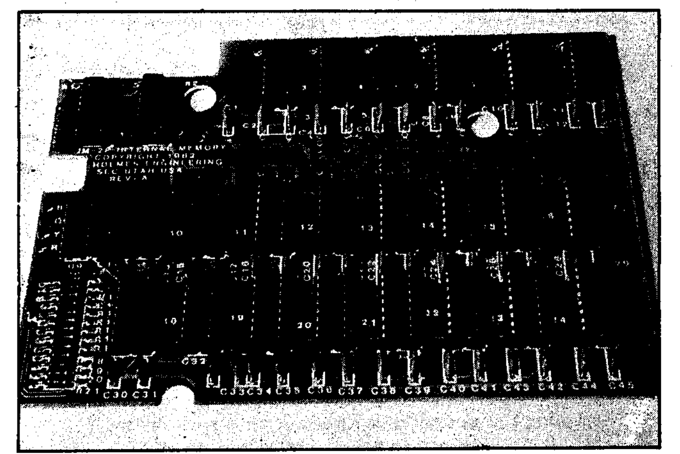
閣下想像中的一部經濟記憶系統，現在已擴展到你的電腦上！—PMC—80記憶擴展單元，其容量可到48K而無須擴張介面。它是插入式的RAM卡，故此不需要用CABLE連接。運行時有可靠的高速度，能避免資料失存和電源過荷的敝端。

IM—I型的擴展單元加上TRS—80MODEL II和MODEL III，可有48K記憶容量，售價爲139.5美元。

IM—2P型加上PMC—80/81，亦可以有48K RAM，售價爲139.5美元。

"INTERNAL MEMORY"是由HOLMES ENGINEERING爲TRS—80和PMC80/81電腦提供其擴展單元的公司。

圖爲"INTERNAL MEMORY"的擴展系統



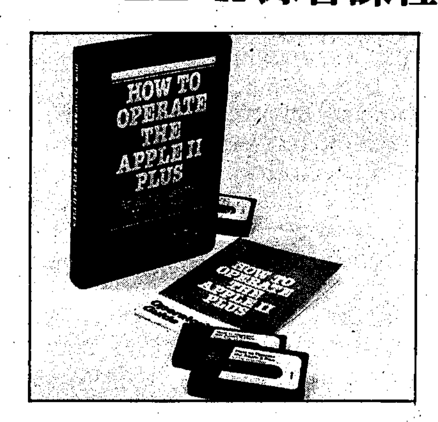
蘋果公事包

ABCOM CORP，16005 SHERMAN WAY，SUITE 105，VAN NUYS，CA 91409。提供一體積較輕、防水的APPLE 電腦公事包。

這部APPLE 板條箱的泡沫塑料套筒，提供震盪和搖擺保護。這堅固的公事包能夠安放在飛機椅下和提供接合器，使你的電腦能與你酒店內的電視機互相連接。

這部APPLE板條箱有一附件袋和公事包，裝載着兩部磁碟機，售價爲92美元。

APPLE II錄音課程

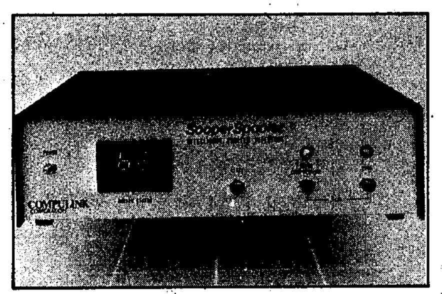


"HOW TO OPERATE THE APPLE II PLUS"是一錄音的迷你課程，適合初學電腦者去學習，內容包括如何運行程序，輸入數據，節省或編組文件，製造後援複製文件，減輕程序等等。

這課程的學費爲50美元，由FLIP TRACK TRAINING TAPES公司主辦，地址爲：

526 N.MAIN ST，BOX 711，GLEN ELLYN，IL 60137，(312)790—1117。

SOOPERSPOOLER是一聰明的印字機介面。再者，可作爲硬件緩衝很多可選擇的軟件格式的特徵是可存取一簡單的BASIC 程序。



基本的機型售價爲394美元。包括一16K BYTES 記憶系統和CENTRONICS 兼容的輸入輸出埠。其他合標準的特徵包括一自容納能源供應，兩位數字 LED 顯示，表示存在於緩衝內的數據總量。

可選擇的附件，包括記憶存貯器擴張到62K，售價爲159美元，和RS—232 串行輸入/輸出埠，售價爲95美元。它能夠利用調制解調器傳輸和由串行到平行翻譯工作。

如欲知悉更詳細資料，可函：

COMPULINK CORP，1840 INDUSTRIAL CIRCLE，LONGMONT，CO 80501，(800)525—6705。

COMPRESSING SOUND

聲音壓縮方法

在一七七五年，森姆爾莊臣會說過：一位作者大部份的時間是用於閱讀上。經過二百年後，此一格言變得更廣泛，意思應爲：人們很多的時間是用於聆聽上，特別在對於收音機或電視。顯然這些時間的大部份全然被浪費一不是被傳遞的訊息荒謬，祇因傳遞的方式無效而已。

早在五十年前，研究人員已發現了人類的理解力在吸收口述訊息比說話快得多。無疑地閱讀比說話快，可能因爲說話包括了相向通信，而文字主要爲單向。在談話中，思考力一面分析所聽到的和同時策劃答案。當閱讀時，沒有機會答覆，因此能夠將資料吸收得更快。祇有當我們反問自己在讀什麽和提出答案時，才使速度減慢。

現今的通信大部份爲口述的和被錄音。在單向通信中聆聽正在代替閱讀。可惜，聽的速度很慢。說話速度爲每分鐘一百一十至一百七十五句，或約爲每秒鐘三點五音節一比我們所能吸收訊息的速度低得多。事實上，研究指出此一慢速妨礙理解；無疑的，會令心不在焉。心理學家指出，即使是有趣的大學演講，學生們百份之六十的時間在發白日夢。

在一九二九年貝爾實驗室之夏維福萊查爾嘗試增加錄音速度來增加資料的傳遞率，但是任何實質的加速就破壞理解力。任何人都知到如將三十三轉的唱片以四十五轉或七十八轉來錄音會產生什麽效果。若以比所錄爲快的速度來唱唱片不但增加唱片拍子的速度（或資料的輸送率）而且也增加訊號的頻率（或音調）。當聲音的音調被提高到相當程度時，揚聲器的聲音有如唐老鴨或艾溫小松鼠的。受過特別訓練的快速報導員約在一九四零年確定了是音調的變化而非資料的高傳送速率破壞理解能力。

在哈佛大學的實驗員發覺到將錄音的小片斷間歇的除去是不會嚴重地減低理解能力（即使資料被除去三份二）。在此，與一九五零年威廉D。加爾維在維珍尼亞大學進行之實驗比較祇不過是一小步。加爾維自然地將錄音帶割去小片斷和將剩餘的接合。這樣錄音帶以三倍速度播出而保有百份之八十聽者之理解力。用此法，音調保持正常（雖然通信是速率增加了）因爲若干資料被拋棄。結果使你能以三倍於正常速度說話但仍然像你的聲音。一位受過資訊理論訓練的人會說語言有很高的適應性，若干清晰度被略去而訊息仍能保持原整。

當然，加爾維之割開及接合技術極爲複雜。幾年前一種電子機械系統被發展出來模擬此一方法而無需割切音帶。第一種系統用一特別錄音機，有四個同帶錄音頭裝在一旋轉鼓上，與錄影帶的錄影頭很相似。

沒有用的部份內容一定要做得很謹愼。顯然地，不可能減去整個字或語句而不破壞內容。而且，也不能將音節弄亂，其出現率約爲每秒十次。因此被放棄片斷祇可以是音節組合單元之一小部份。這些音位一組成說話之基音一長短約千份之一七五秒。事實上，被放棄之片斷時間不能長於萬份至二萬份之一秒。因此，很多小片斷需要放棄而每一斷都需要接駁。接合小片斷要很小心。若然片斷間的訊號強度相差太大，會產生突然的失眞。

今天之電子及電腦技術已代替了五零和六零年代的電子機械時間壓縮器，而且靈活性更大。本質上，純電子技術與早期的機械縮壓技術是很相似的。若干新式的時間縮壓系統使用模擬技術而其他用數字技術，但全部需要將訊號分成小片斷，比那些保留或放棄片斷更短。此動作名爲訊號樣板化。

資訊理論指出任何訊號在規定的間隔取出，可從樣板中重建。重建是與連點製圖相似。濾波電路用作樣板中之橋樑如同鉛筆將各點組合。很多的點才能產生詳細的圖像；同樣地，需要很多樣板來保持自然，沒有失眞的聲音或音樂。

若干說話壓縮系統工作於模擬式訊號（雖然被樣板化）。而更精密的裝置則使用電腦技術。每個模擬樣板被轉爲一數字〝句〞，一組字句可將樣板重建。這些數字語句被順序輸入一數字式隨意記憶內。微處理機（電腦）根據程序接觸記憶及取出資料。電腦決定以那一斷接得最適合，多少被放棄及多少被保存。在組合好那些數位及片斷變成數字語句再變爲模擬樣板後再被濾波。由於電腦祇被其記憶容量和工作速度所限制，功能幾乎是無限的，與及壓

縮系統可立即再入程序來壓縮或擴展時間，改變保存及放棄之片斷（同時保持它們之間之壓縮一擴展比例），或改變保存一放棄比例來改變時間因素。

由於合理化價錢的系統的出現，學生們可以迅速重溫一場演講，盲人可以用閱讀的速度來聽一本書，秘書們可以校好適當的重播速度來配合打字速度。由於所需的積體電路之出現令到可變說話控制公司能生產出一實用，合理低價的系統（二百美元以下）。

使用數字技術之專業時間壓縮系統相當昂貴（LEXICON值八千五百美元而INTEGRATED SOUND SYSTEMS值六千至八千美元），但高價不會影響到它們的使用。電台將六十三秒商業廣播縮爲六十秒來配合新聞廣播的準確時間（及加速慢講的記者之說話速度），及在節目間加插音樂。

變速攝影機或錄影機與時間壓縮之同步運用已在電視台使用。電影用作配合放影時間，新聞廣播增加流量，與及將球賽和實地播送變得比現場更爲緊張。錄影帶翻錄將兩小時十分的電影壓縮爲兩小時，與及廣告社能創新廣告製作不但能令觀衆接受而且可留下更佳印象。如果驚奇廣告中動得和說得這麽快，記着一那是壓縮技術。

模擬及數字式語言壓縮系統如何工作。

用電子方法來作語言壓縮或擴張是依靠訊號樣板化，那裏之電路以短暫時間〝視察〞該訊號及將其值儲存，通常利用電壓值。在數字系統中，另一電路名爲模擬至數字轉化器。以數字代表電壓，它可以用一連串的脉冲來表示及可由一些電腦式電路處理。在模擬式系統則直接使用電壓樣板。每一模擬樣板用適當電壓寫進一儲存單元，而可以由另一電路讀出或感應。此一電路可將樣板轉移至另一儲存單元。此工作繼續至用到最後之單元。一組單元以這種方式連接名爲模擬移位寄存器，或吊桶隊。

有如在火災中救火的吊桶輸送，經過移位寄存器之電壓需要同步的。將電壓寫入儲存單元而又未將原有資料讀出將會毀壞資料。用作同步轉移，一個名爲時鐘電路移位寄存器輸入脉冲。事實上，一連串之樣板一現在代表出現在寄存器輸入區之信號一根據時鐘所設立之節奏而通過該元件。

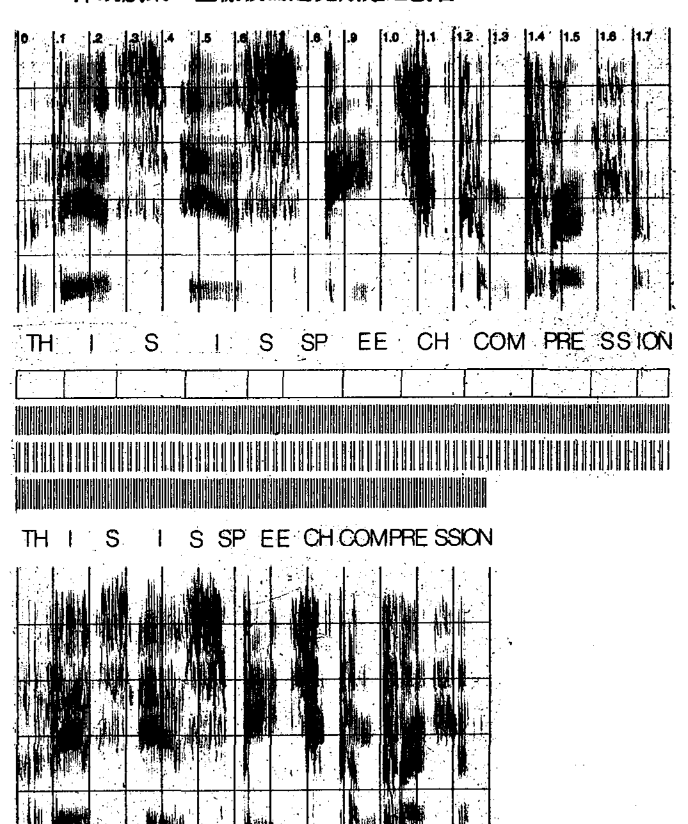
不論在模擬或數字系統，當目標爲壓縮說話，訊號是取自快速之回轉錄音一事實上，在能聽到之輸出速度。爲避免訊號之音調提升該電路將若干樣板放棄。

用作訊號擴展一減慢說話速度一音帶也是用所需之輸出速度播出。壓縮器將若干樣板使用兩次，因此音調提升至正常。

絕無理由需要音調及信號速度保持固定關係。音調可以調升或降低而不必理會訊號速度，巨人、松鼠、或怪物之聲音的特殊效果因此而被發展出來。名歌星用該技術將他們的聲音由中音變爲高音或相反一有時還加上一些特別的效果。

當樣板被取出或加入訊號中，是需要接合因爲它們原本不是連續的。若然兩個連續樣板之間的差別太大，會產生失眞。改善之法爲將兩個樣板以較接近值的另外兩個來代替以減少差別。這方法很多數是可行的，但精密的電路能做得更好。

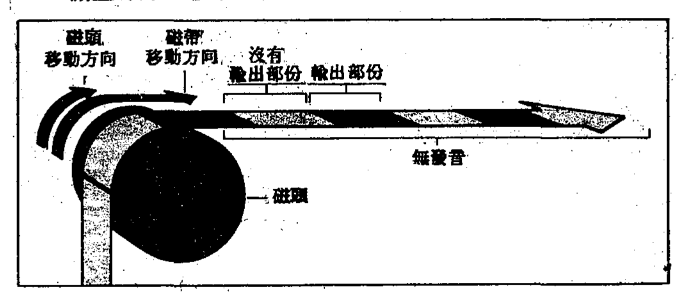
該原理容許訊號作樣板時不必使樣板達至一致，雖然通常這是最爲方便的。最重要是在訊號最高頻處同時作兩次樣板。這樣若要將電話中的說話作樣板而在三千赫中刪去，可以將資料用六千份之一秒錄一次或每三千份之一秒錄兩次及保留訊號原整。使用此一寬容度，電腦線路可選擇或放棄一些樣板而避免艱難之接合。



訊號模擬樣版

在圖一中，短語的速度如圖中所示的濃縮，在說話時，其音波形成的波音圖。它會持續0．010到0．020秒內切成片段而要短過發音時間。

在棄掉和接合的過程中，其進行時是利用一轉動的磁帶頭而具備有四個檢查的隙口。當磁帶移動在高速時，這檢查罅隙亦移動在同一方向。正如圖二所示。



AMAMAM

立體聲廣播系統介紹

調幅立體聲廣播的來臨

在立體聲廣播方面，FM調頻立體聲已出現了多年，但是由於方向性，障礙物之阻擋，惡劣天氣之衰減等接收問題，令到了某些地區接收不到或效果不佳，但是，如使用調幅廣播就可減輕這些不良效果。市面上至今還未見有售調幅立體聲收音機是什麼理由呢？

事實上，技術上的問題早已解決，留下來的祇是標準系統的選擇問題。

說回在一九七七年，美國聯邦通信委員會收到了五家電子公司落標競投作為標準系統的供應者，那就是比拉爾（BELAR），哈利斯（HARRIS）、汗／赫宙庭（KAHN／HAZELTINE）、麼托羅拉（MOTOROLA）與及（MAGNAVOX）。

當一九八零年聯邦通訊委員會選擇了MAGNAVOX系統後，其他四家公司立刻上訴，數月後，該委員會收回了所發出之宣報，代之以重審資料，至今仍然懸而未決。

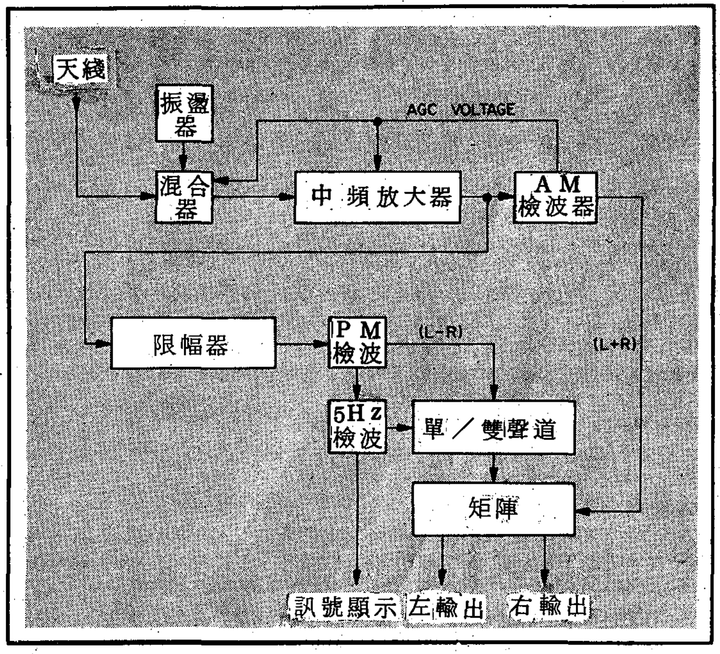
由於未能選出標準系統，聯邦通訊委員會現已批準美國的調幅廣播電台使用任何一種系統來廣播。在這種各自為政的情況下，聽衆祇有將選擇權交給家電和汽車立體聲音響製造商來決定。如果這五種系統在工作上是相近的話，祇要在單一線路上作一些修改是不會令到收音機的成本增加很多的，但是，這五種系統是節然不同的。

所以製造商們所採的決定會是：

1，選擇製造單一系統之收音機。

2，製造擁有數個解碼線路而可由收聽者自行選擇或會根據接收到的電波而自動選擇解碼線路的收音機或接收器。

此外，產生另一問題的是等待電台選擇何種廣播系統後才生產調幅立體聲收音機而同時電台等待製造商推出了產品後才決定所用之系統，成為製造商等電台而電台又等製造商的局面。



此圖是接收機的方框圖

調幅立體聲廣播是不會令到現有收音機會落伍，不過工業界估計，如果能定出一個標準系統的話，到一九八三年將會售出五百萬部調幅立體聲收音機。現時，起碼已有六家汽車音響製造廠已製成了調幅立體聲收音機的試驗型。

在音響製造商方面，部份已決定了其所用之系統，其中利迅半導體公司，世界上著名的積體電路製造商，自一九七六年已打入調幅立體聲的市場，在一九八零年，當聯邦通訊委員會選擇了MAGNAVOX系統時已準備推出其調幅立體聲解碼器的集成電路，現時，已發展到單片積體電路（LM1981）其基本為MAGNAVOX系統而設計，但作了一些修改（更換外接元件）後，也可用在其他至少兩種系統上（麼托羅拉及比拉爾），及可能增減

一些其他元件後也可用在其他兩個系統之上（哈利斯及汗／赫宙庭）。顯然，這晶片是可用在至少另外兩個系統上，利迅半導體公司是選擇MAGNAVOX系統爲對像。日本PIONEER先鋒公司也是支持MAGNAVOX系統的。

山水亦製成了一試驗型的接收器，是能自動對MAGNAVOX，哈利斯或汗／赫宙庭系統進行檢波和解碼，但該公司現時未決定支持任何一個系統。

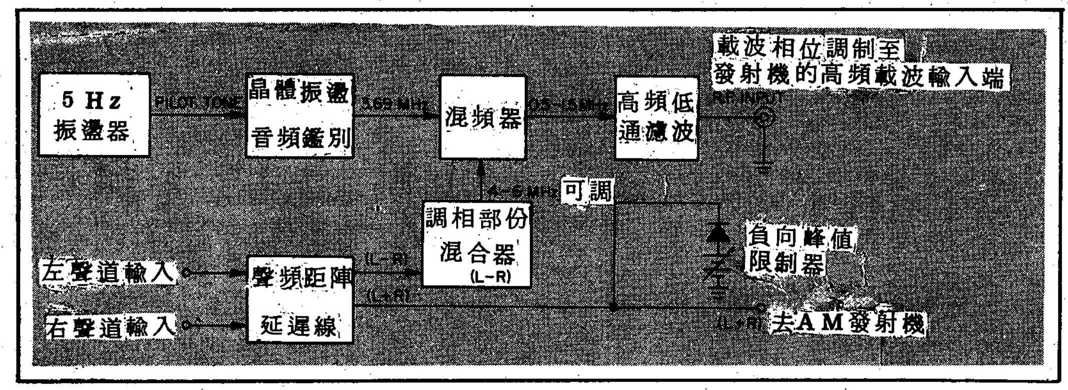
福特汽車公司屬下之福特航太空與通訊公司之生產汽車收音機部門支持使用MAGNAVOX或比拉爾系統，該公司認爲麼托羅拉，哈利斯及汗／赫宙庭系統的價錢爲比拉爾或MAGNAVOX系統的雙倍，但沒有顯著的優勝。

哈利斯則積極推廣自己的系統，它是唯一生產發射機的公司，去年供應了超過一半全美國的廣播電台所用的調幅及調頻發射機。它宣稱收到了一百五十張訂單來供應發射系統給各調幅立體聲電台。

在日本的情況也是一樣，該國政府也不能選出標準的調幅立體聲系統。三洋、松下、天樂、愛華、新力、聲寶、星牌、日本哥林比亞、雅佳等正在等候更多的資料來決定用那種系統。

最後，技術性能最佳之系統亦不一定會被選擇，因爲商業推廣和價錢在系統選擇方面比性能優異更具影響力，況且，五個系統中那個最佳也引起爭論，包括在聯邦通訊委員會內。

註：尚有AM立體聲資料刊登於現代電子 1980年第8期。



此圖是立體聲發射機之方框圖

多用途的儲存程式

筆者的朋友曾在某處借來了一份文字處理說明書，他強調APPLE的"MAKE TEXT"程式可以用作郵寄標貼（MATTLING LABELS），所以一天筆者和幾位同好一起運算那個程式，因爲筆者不相信他朋友的說話。

經過一番思考和研究後，筆者便把它改成以下的程式：

在列表中列出了七項特徵：

1）製造文字（MAKE TEXT）
2）提取文字（RETRIEVE TEXT）
3）表現文字，藉此校對（TO CHECK EUTRIES）
4）貯存文字（SAVE TEXT）
5）改變文字（CHANGE TEXT）
6）目錄（CATALUG）
7）印出文字（PRINT TEXT）

很意外地，這個細小的程式用途實在很廣，甚至可媲美市面職業化的軟件，你可以用它來紀錄僱員的地址，名字和其他資料，可以印出你需要的資料。筆者相信讀者們必定能夠找出其他更多的用處。

```
10 REM  **  PRIMITIVE WORD  **
20 REM  **  PROCESSOR  AND  **
30 REM  **  DARA FILE (MAY  **
40 REM  **  BE  USED AS  A  **
50 REM  **  MAILING  LIST)  **
60 REM  BY HENRY KONG
70 DIM A$(100)
80 D$ = CHR$ (4)
90 REM  MENU
100 PRINT  CHR$ (7)
110 HOME : PRINT : PRINT : PRINT
120 HTAB (15): INVERSE : FLASH : PRINT " M E N U ": NORMAL
130 PRINT : PRINT "(THIS PROGRAM MAY BE USED AS A WORD PROCESSOR, MAILING LIST O
R DATA FILE)
140 PRINT : PRINT : PRINT TAB( 5)"1) WRITE TEXT
150 PRINT : PRINT  TAB( 5)"2) RECALL TEXT FROM DISK
160 PRINT : PRINT  TAB( 5)"3) DISPLAY TEXT ON SCREEN
170 PRINT : PRINT  TAB( 5)"4) CORRECT AND UPDATE TEXT
180 PRINT : PRINT  TAB( 5)"5) SAVE TEXT ON DISK
```

```
190 PRINT : PRINT  TAB( 5)"6] CATALOG OF FILES
200 PRINT : PRINT  TAB( 5)"7] PRINT TEXT
210 PRINT : PRINT  TAB( 5)"8] QUIT
220 PRINT : PRINT : PRINT TAB( 12)"SELECTION (1-8)";: INPUT X: HOME
230 ON X GOTO 240,380,590,720,960,1110,1190,1400
240 REM  MAKE TEXT
250 PRINT : PRINT : PRINT
260 HTAB (9): INVERSE : FLASH : PRINT" WRITE TEXT ": NORMAL
270 I = 0
280 PRINT "ENTRE YOUR TEXT A LINE AT A TIME...
290 PRINT "KEEP YOUR LINES WITHIN 40 CHARACTERS IF
300 PRINT "POSSSIBLE (ALTHOUGH UP TO 200+ IS OK)
310 PRINT :I = I + 1
320 PRINT "TYPE '*' TO QUIT ENTRE LINE #";I
330 PRINT "            # OF CHARACTER COUNTER
340 PRINT "1........10.........20.........30.........40
350 INPUT "";A$(I)
360 IF A$(I) (  ) "$" GOTO 310
370 GOTO 100
380 REM RETRIEVE TEXT
390 PRINT : PRINT : PRINT
400 HTAB (6): INVERSE : FLASH : PRINT " PETRIEVE TEXT FROM DISK": NORMAL
410 PRINT : PRINT : PRINT
420 PRINT "THIS PROGRAM RETRIEVES TEXT FILES"
430 PRINT "CREATED BY THE 'CREATE TEXT' PROGRAM."
440 PRINT "MON C,I,O IS IN EFFECT"
450 PRINT
460 INPUT "NAME OF TEXT FILE?";Z$
470 PRINT D$;"MON C,I,O"
480 PRINT
490 PRINT D$;"OPEN ";Z$
500 PRINT D$;"READ ";Z$
510 INPUT I
520 FOR J = 1 TO I
530 INPUT A$(J)
540 NEXT J
550 PRINT D$;"CLOSE ";Z$
560 PRINT : PRINT : PRINT "HIT SPACE BAR TO  RETURN TO MEUN": GET ZZ$
580 GOTO 100
590 REM LIST TEXT
600 PRINT : PRINT : PRINT
610 HTAB (13): INVERSE : FLACH : PRINT " LIST TEXT ": NOEMAL : PRINT : PRINT
620 FOR X = 1 TO 2000: NEXT
630 FOR J = 1 TO I - 1
640 IF J / 20 = INT (J /20) THEN GOSUB 700
650 PRINT A$(J)
660 NEXT
670 GOSUB 700
680 PRINT : INVERSE : PRINT " FINISHED!! RETURNING TO MENU NOW..."
690 FOR X =  1 TO 1500:NEXT : NORMAL : GOTO 100
700 PRINT : PRINT "HIT ";: INVERSE : PRINT " RETURN ";: NORMAL : PRINT " TO COUT
INUE LISTING
710 GET ZZ$: PRINT : PRINT : RETURN
720 REM CORRECT/UPDATE TEXT
730 PRINT : PRINT : PRINT
740 HTAB (8): INVERSE : FLASH : PRINT " CORRECT /  UPDATE TEXT ": NORMAL
750 PRINT : PRINT : PRINT
760 FOR X = 1 TO 2000: NEXT
770 HOME
780 FOR J = 1 TO I - 1
790 IF J /16 = INT (J  / 16) THEN GOSUB 950
800 PRINT J">"A$(J)
810 NEXT
820 GOSUB 850
830 PRINT : INVERSE : PRINT "FINISHED!!  RETURNING TO MENU NOW..."
840 FOR X = 1 TO 1500: NEXT : NORMAL : GOTO 100
850 VTAB 21: PRINT "......................................
860 VTAB 22: PRINT "DO YOU HAVE ANY CHANGES ON THIS PAGE";: INPUT Y$
870 IF LEFT$ (Y$,1) = "Y" THEN 950
880 VTAB 22: PRINT "                                            "
890 VTAB 22: INPUT "WHICH LINE # TO CHANGE: ";: INPUT X
900 VTAB 22: PRINT "                                            "
910 VTAB 22: INPUT "ENTRE NEW LINE NOW: ";A$(X)
920 VTAB 22: PRINT "                                            "
930 VTAB 22: INPUT "ANY MORE CHANNGES ON THIS PAGE? ";Y$
940 IF  LEFT$ (Y$,1) = "Y" THEN VTAB 22: PRINT "
    ": GOTO 890
950 HOME : RETURN
960 REM  SAVE TEXT
970 PRINT : PRINT : PRINT
980 HTAB (9): INVERSE :  FLASH : PRINT " SAVE DATA ON DISK ": NORMAL
990 PRINT :  PRINT : PRINT
1000 INPUT "WHAT FILE NAME? ";N$
1010 PRINT D$;"OPEN ";N$
1020 PRINT D$"DELETE ";N$
1030 PRINT D$;"OPEN ";N$
1040 PRINT D$;"WRITE ";N$
1050 PRINT I -1
1060 FOR J = 1 TO I - 1
1070 : PRINT A$(J)
1080 NEXT J
1090 PRINT D$;"CLOSE ";N$
1100 GOTO 100
1110 REM  CATALOG
1120 PRINT : PRINT : PRINT
1130 HTAB (15): INVERSE : FLASH : PRINT " CATALOG ": NORMAL
```

apples apples apples apples apples

DOS3.2改3.3的方法

SINGLE-DRIVE

很多人常常談論怎樣把DOS 3・2的磁碟變成DOS 3・3，但從沒有人把這項工作實踐。

筆者從一本雜誌中看到一個單磁碟的抄寫程式（它是用INTEGER BASIC寫成的）筆者把它其中的一部份取了出來和MASTER磁碟中的MUFFIN聯合運用，產生一個單磁碟機3・2轉爲3・3的程式。它不像MUFFIN般，每次處理一個檔案後，便要把磁碟換去一部48K的APPLE Ⅱ具備足夠RAM每一次能夠輸送100個頁區，意思是指一張3・2的磁碟可以在五次內變成3・3。

應用這個程式時（必須把程式貯存在格式3・3的磁碟內）。碟內除包括了這個程式外，還必須將MASTER碟內的MUFFIN貯存在碟中。接受3・3檔案的磁碟必須在使用前INIT（INITIALIZED）。否則磁碟必須空白。任何在磁碟上的檔案在抄閱的時候都可能被刪去。開始運算以下程式時，程式便會提示使用者在適當的時候放入磁碟。

行0設置LOMEM致8460，爲MUFFIN保留空間，在行500時載入的。注意LOMEM必須在行1定出變數時而設置之。

在行1000前的評語爲了速度的關係，把它們移到最後。

行1067，4035和4037把DOS的指針（POINTERS）改至適當3・2或3・3的RWTS（READ OR WRITE A TRACK AND SECTOR）常式（ROUTINE）。這便是基本的轉向步驟，容許DOS 3・3讀取13頁區（3・2）的磁碟。因爲只有3・2磁碟被輸入，行1、1080和2040－2095將被改變，消去VER這個變數。而且，我們不希望把3・2DOS抄進3・3的磁碟內。行2010和2040把第三軌至34軌輸送，剩下軌0－2。

行2100－2120把在VTOC（VOLUME TABLE OF CONTENTS）的3・2BIT MAPS改變。在行4070的GOSUB跳至一個新副程式（行6000－6110）執行完畢後，向行動便完畢了。（行6000－6110）是把VTOC重寫和重新校正索引（DIRECTORY）的連接。然而，行4070容許重新開始一個多磁碟的轉向（只要用者需要）

注意一張沒有標準DOS索引（在軌17）的磁碟或多數不能抄寫（UNCOPYABLE）的磁碟要改成3・3，則不能應用以下的程式。

```
>LIST

   0 POKE 74,8460 MOD 256: POKE
     75,8460/256: POKE 204, PEEK
     (74): POKE 205, PEEK (75): REM
     LOMEM:8460
   1 A=B=PTR=LOC=RWTS=IBTRK=IBSECT=
     IBBUFP=REP=REPS=CMD=TBL=IOB=
     TRK=SEC=BYTE1=BYTE2=OLDPTR=
     BITMAP=BUFLO=BUFHI=IBVOL=IBCMD=
     0
 500 PRINT "BLOAD MUFFIN": REM CTRL-D

1000 TEXT : CALL -936: VTAB 4: TAB
     11: PRINT "SINGLE DRIVE CONVERT"
     : TAB (16): PRINT "3.2 -> 3.3"
     : VTAB 7: TAB 11: PRINT "BY DANA
      J. SCHWARTZ"
1005 TAB 11: PRINT "WASHINGTON APPLE
     PI": VTAB 12: TAB 16: PRINT
     "BASED ON": TAB 11: PRINT "SINGL
     E DRIVE COPY.3": TAB 13: PRINT
     "BY STEVE ADAMS"
1010 VTAB 20: PRINT "INSERT THE DISK(
     3.2) YOU WISH TO CONVERTAND GENT
     LY TOUCH RETURN TO BEGIN.":
      GOSUB 5010
1020 REM
1021 REM *** FIND THE IOB ***
1022 REM
1030 A= PEEK (77): IF A>94 THEN
     A=A-256:IOB=(A+33)*256+231
1040 REM
1041 REM *** LOAD CONTROLLING ***
1042 REM *** SUBROUTINE IN    ***
1043 REM *** PAGE 0           ***
1044 REM
1050 POKE 0,169: POKE 1,IOB/256+
     255*(IOB<0): POKE 2,160: POKE
```

```
     3,232: POKE 4,32: POKE 5,217
     : POKE 6,3: POKE 7,96
1060 REM
1061 REM *** LOAD THE VTOC INTO ***
1062 REM *** MEMORY STARTING    ***
1063 REM *** AT $02D0           ***
1064 REM
1065 IBVOL=IOB+4:IBTRK=IOB+5:IBSECT=
     IOB+6:IBBUFP=IOB+10:IBCMD=IOB+
     13
1067 POKE -17152,76: POKE -17151
     ,0: POKE -17150,30: REM MUFFIN
     RWTS
1070 POKE IBVOL,0: POKE IBTRK,17
     : POKE IBSECT,0: POKE IBBUFP-
     1,208: POKE IBBUFP,2: POKE
     IBCMD,1: CALL RWTS
1080 BITMAP=776
2000 REM
2001 REM *** THE TABLE STARTS ***
2002 REM *** AT "TBL"         ***
2003 REM
2010 TBL= PEEK (204)+ PEEK (205)
     *256+1:PTR=TBL
2020 VTAB 10: TAB 15: PRINT "I'M THIN
     KING"
2030 REM
2031 REM *** READ "BIT MAPS" IN ***
2032 REM *** THE VTOC AND CON-  ***
2033 REM *** VERT TO BINARY     ***
2034 REM
2040 FOR TRK=3 TO 34:BYTE1=BITMAP+
     TRK*4:BYTE2=BYTE1+1:SEC=12
2050 A= PEEK (BYTE1): IF A#255 THEN
     2070:SEC=4
2060 A= PEEK (BYTE2): IF A=248 THEN
     2100
2070 B=A/128:A=A-B*128: GOSUB 3020
     :B=A/64:A=A-B*64: GOSUB 3020
     :B=A/32:A=A-B*32: GOSUB 3020
     :B=A/16:A=A-B*16: GOSUB 3020
     :B=A/8:A=A-B*8: GOSUB 3020
2080 IF SEC<0 THEN 2100
2090 B=A/4:A=A-B*4: GOSUB 3020:B=
     A/2:A=A-B*2: GOSUB 3020:B=A:
     GOSUB 3020
2095 GOTO 2060
2100 IF TRK<3 THEN 2120
2110 A= PEEK (BYTE1):B= PEEK (BYTE2)
     : POKE BYTE1,A/8+224: POKE
     BYTE2,B/8+(A MOD 8)*32: REM CON-
     VERT BITMAP
2120 NEXT TRK: GOTO 4010
3000 REM
3001 REM *** IF THE TRACK BIT  ***
3002 REM *** MAP INDICATES AN  ***
3003 REM *** IN-USE SECTOR,    ***
3004 REM *** POKE TRK & SEC    ***
3005 REM *** INTO THE TABLE    ***
3006 REM *** STARTING AT "TBL" ***
3007 REM
3020 IF B THEN 3030: POKE PTR,TRK:
     POKE PTR+1,SEC:PTR=PTR+2
3030 SEC=SEC-1: RETURN
4000 REM
4001 REM **** COPY ****
4002 REM
4010 BUFLO=(PTR) MOD 256:BUFHI=(
     PTR)/256: POKE IBBUFP-1,BUFLO:
     POKE IBBUFP,BUFHI
4020 REPS= PEEK (203)-BUFHI-( PEEK
     (202)<BUFLO):OLDPTR=TBL:TBL=
     PTR
4030 FOR CMD=1 TO 2: CALL -936: IF
     CMD=1 THEN PRINT "READING":
     IF CMD=2 THEN PRINT "WRITING"
     : POKE IBCMD,CMD:LOC=BUFHI:
     PTR=OLDPTR
4035 POKE -17152,76: POKE -17151
     ,0: POKE -17150,30: REM MUFFIN
     RWTS
4037 IF CMD=1 THEN 4040: POKE -17152
     ,132: POKE -17151,72: POKE
     -17150,133: REM 3.3 RWTS
4040 FOR REP=1 TO REPS: POKE IBTRK,
     PEEK (PTR): POKE IBSECT, PEEK
     (PTR+1): POKE IBBUFP,LOC
4045 VTAB 3: PRINT "TRACK="; PEEK
     (IBTRK);: TAB 12: PRINT "SEC="
     ; PEEK (IBSECT);" "
4050 CALL RWTS
4060 LOC=LOC+1:PTR=PTR+2: IF PTR#
     TBL THEN 4080
4070 IF CMD=1 THEN 4090: GOSUB 6010
     : CALL -936: PRINT "FINISHED"
     : PRINT : INPUT "ANOTHER DISK (1
     =Y/0=N)",A: IF A=1 THEN 1000
     : END
4080 NEXT REP
4090 FOR A=1 TO 1000: NEXT A: CALL
     -936: VTAB 5: PRINT "INSERT THE
     ";: IF CMD=1 THEN PRINT "DUPLICA
     TE(3.3)";
4100 IF CMD=2 THEN PRINT "ORIGINAL(3.
     2)";: PRINT " AND HIT RETURN"
     : GOSUB 5010
4110 NEXT CMD:OLDPTR=PTR: GOTO 4030
5000 REM
5001 REM *** WAIT FOR 'RETURN' ***
5002 REM
5010 POKE -16368,0
5020 IF PEEK (-16384)#141 THEN 5020
     : POKE -16368,0: CALL -936:
     RETURN
6000 REM
6001 REM *** HANDLE TRK 17 ***
6002 REM
6010 CALL -936: VTAB 3: PRINT "REVISI
     NG VTOC & DIR"
6020 POKE 722,15: POKE 723,3: POKE
     726,254: POKE 773,16: POKE
     844,0
6030 POKE IBTRK,17: POKE IBSECT,
     0: POKE IBBUFP-1,208: POKE
     IBBUFP,2: CALL RWTS: REM REWRITE
     VTOC
6040 FOR A=12 TO 1 STEP -1
6050 POKE IBSECT,A: POKE IBCMD,1
     : CALL RWTS: REM READ DIR
6060 POKE 721,17: POKE 722,A+2: POKE
     IBSECT,A+3: POKE IBCMD,2: CALL
     RWTS: REM SHIFT & REWRITE
6070 NEXT A: FOR A=720 TO 975: POKE
     A,0: NEXT A: REM DO LAST 3
6080 POKE IBSECT,1: CALL RWTS
6090 POKE 721,17: POKE 722,1: POKE
     IBSECT,2: CALL RWTS
6100 POKE 722,2: POKE IBSECT,3: CALL
     RWTS
6110 RETURN
9000 REM **************************
9010 REM *                        *
9020 REM *   SINGLE DRIVE CONVERT *
9030 REM *       3.2 -> 3.3       *
9040 REM *                        *
9050 REM **************************
```



AM Stereo Orders Start

AM 立體聲介紹

MAGNAVOX調幅立體聲廣播系統與其解碼器

甚麼是調幅立體聲？

調幅立體聲是將調幅波廣播產生三度空間的音嚮效果。調幅一中頻訊號被分爲左和右聲道，使接收機從揚聲器中發出多方位之聲音效應。

爲甚麼需要調幅立體聲廣播

調頻立體聲廣播已使用多年，同時證明了能對本區之聽衆提供優質之立體聲廣播節目，爲何在最近幾年人們這樣熱烈地發展調幅立體聲廣播？這裏有兩個主要的理由：

一，近年來，調幅廣播電台之聽衆續漸減少，因爲錄音機和盒帶機之普遍化與及調頻廣播電台之立體聲廣播（大部份已採用）。

二、調頻立體聲廣播之基本缺點。就是調頻廣播之輸送功率衹有調幅廣播五分之一，令到距離發射台較遠的地區接收不佳和產生多路反射效應（接收天線同時收到另外一個時間延遲訊號，令到接收訊號變形）和需要直線輸送。這些缺點在使用汽車收音機時更顯著。

在調幅立體聲廣播方面一定能提供更大的立體聲廣播範圍和消除多路效應，尤其是在行駛的接收。

聯邦通訊委員會之最新決定

早在一九八零年，MAGNAVOX系統已被美國聯邦通信委員會選爲工業界調幅立體聲的標準。但，不久後，其他四個系統的代表進行上訴，結果推翻了原先決定，改爲重審這五個系統。在一九八二年三月十八日，該委員會仍認爲MAGNAVOX系統最佳，但批準了自一九八二年三月四日起，美國的調幅立體聲廣播電台是可以用任何一個系統來作廣播。不過，收音機製造商們相信在不久將來，根據市場需求會求出一個系統來作爲工業標準。

MAGNAVOX 調幅立體聲系統

在參與調幅立體聲系統競投之系統中，不少被淘汰了，最後剩下的如

一、調幅一調相
二、調幅一調頻
三、獨立邊帶
四、正交邊帶

第一個系統調幅一調相，是由MAGNAVOX提出的。它提供了如下的優點：

1、簡單
2、低失眞
3、與單聲道匹配
4、分音良好
5、頻率反應佳
6、低噪音

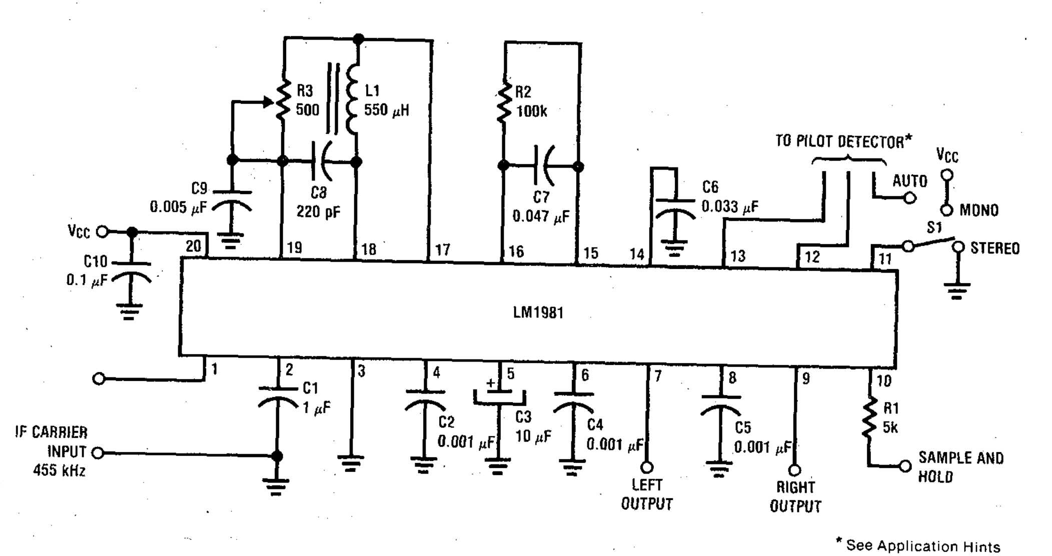


圖1是LM1981的外接元件圖

現時，其他系統也擁有若干以上的優點，但是，除了MAGNAVOX系統外，其他系統是未能擁有全部的。

MAGNAVOX系統由三組訊號所組成。左與右聲頻訊號之和（L＋R）用作載波之調幅。左與右訊號之差（L－R）作爲同一載波之調相至峰值＋或－之一個弧度。加上、一個五赫之立體聲認別音頻，用於＋或－四個弧度之峰值調變。該五赫指導音頻令接收器自動開關單聲道或立體聲道之線路和那常見的立體聲指示燈。

一部標準的調幅／調頻立體聲接收器是可以接收調幅立體聲廣播的，衹要加上一個調幅立體聲解碼器就行。基本上，整個調幅／調頻載波由一個超外差前級轉爲中頻。在中頻放大器輸出訊號分爲兩個方向。一個典型的波封檢波器可用作取出調幅資料及檢出載波水平作爲自動增益控制之用。同時該中頻放大器輸出衹用作消除波幅之調變，及一個整流器提供（L－R）聲頻訊號和五赫立體聲認別音調。調幅及調相波道一起用作矩陣來推動左聲道和右聲道之聲頻功率放大器。

至今，利迅半導體公司（NATIONAL SEMICONDUCTOR CORP），是唯一有調幅立體聲解碼晶片在市場出售之美國半導體製造商。

LM1981爲一單片積體電路被設計作爲立體聲訊息之解碼用，那是用作調幅立體聲廣播的載波之調幅及調相。它可接受455千赫（或262千赫）中頻輸出及調幅檢波（L＋R）單聲道訊號；限制，整流及調節（L－R）立體聲訊號差；及將這些訊號組合於一個適當之矩陣來作左及右頻道的音頻輸出。其他特性包括一個超相檢波器，立體聲指導音頻之輸出，立體聲／單聲道混合功能，輸出樣本及保持綫路和一個內調參考電壓。

結論

MAGNAVOX調幅立體聲系統是能匹配單聲道的。在立體聲接收型式之聲道分隔是高於二十分貝。沒有固有頻率反應的限制。此系統能滿足廣大公衆需求。在實驗室測試時功能良好。相信該系統會受發燒友們及廣播界之歡迎。

NS半導體之調幅立體聲解碼器（LM1981）

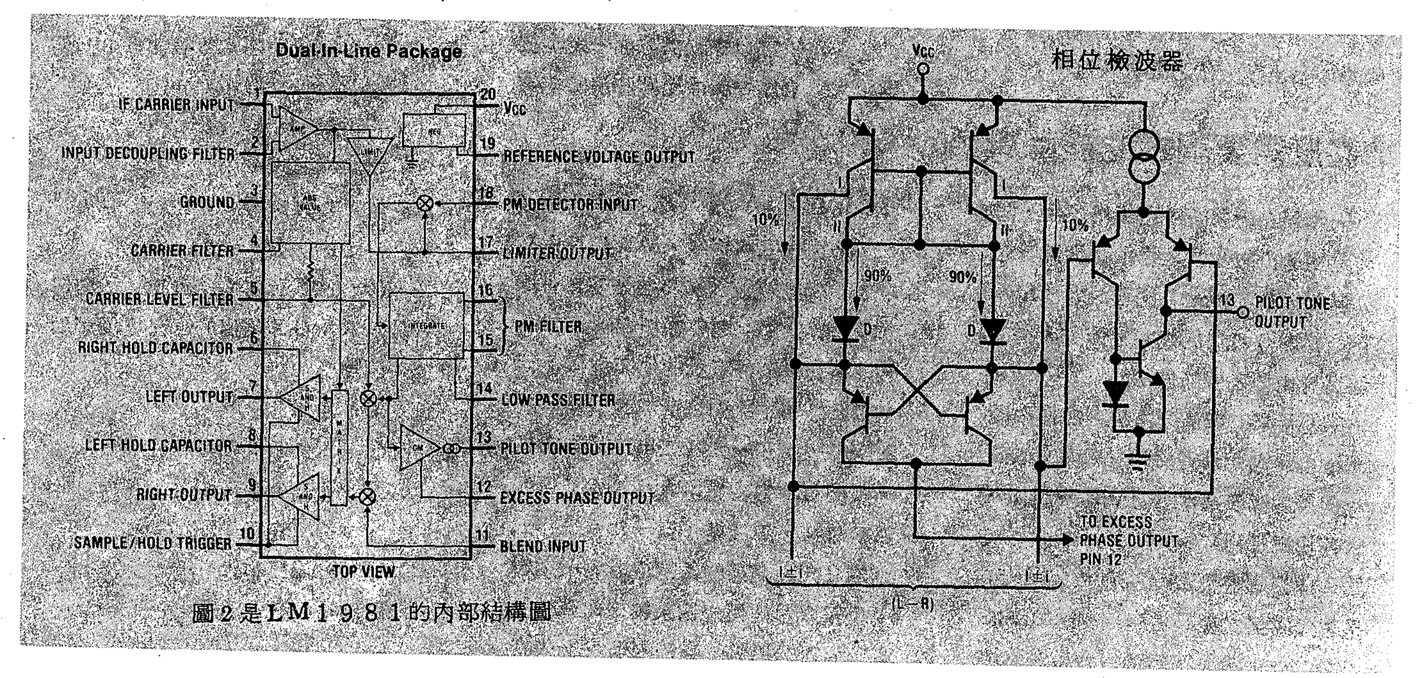
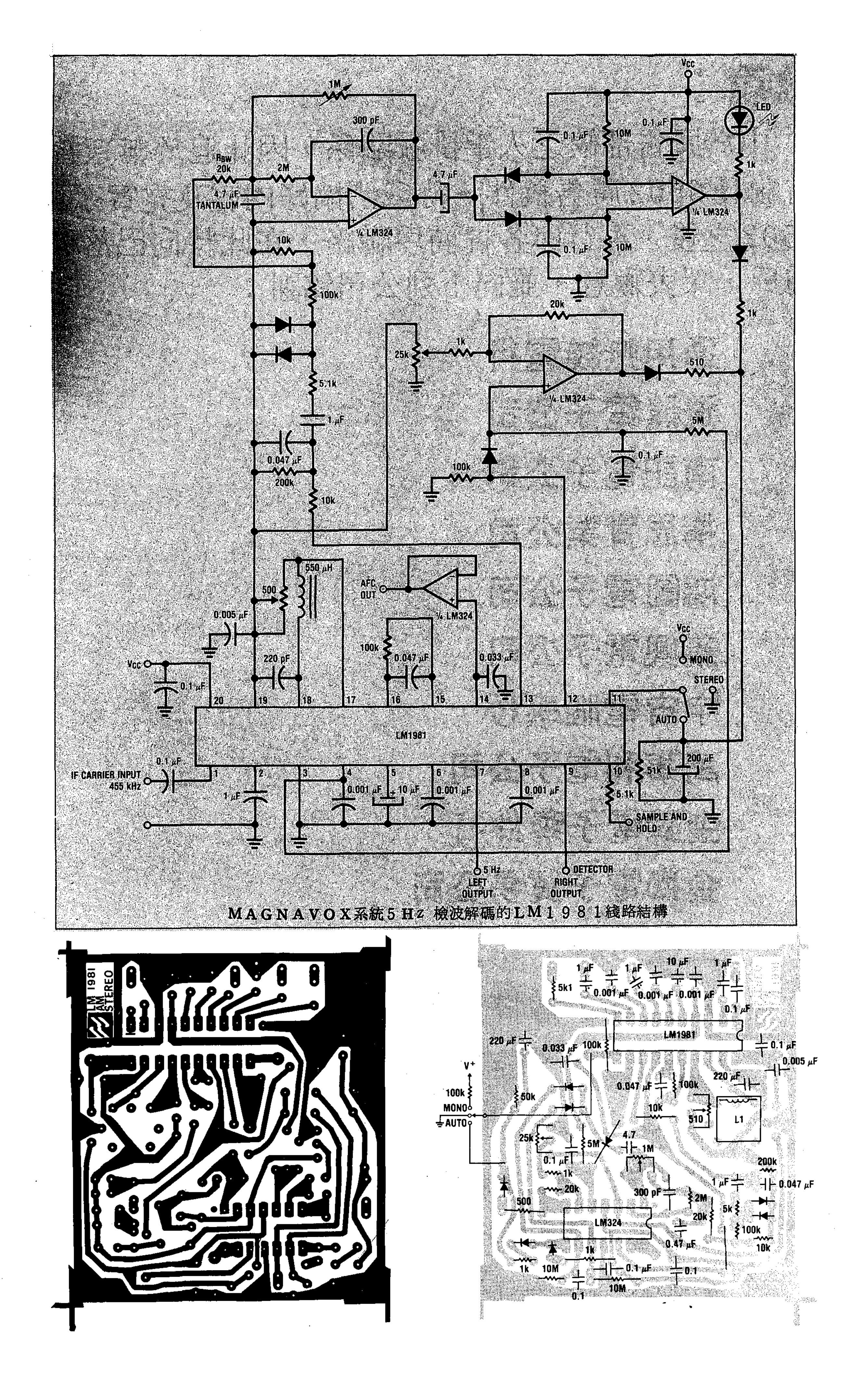


圖2是LM1981的內部結構圖



MAGNAVOX系統5Hz 檢波解碼的LM1981線路結構

致歉篇：

由於廣告部缺乏人手排版關係，因此迫於無奈暫停有關IE6502廣告稿件，待我們搞好內容及充實了IE 6502之後，再行與各貴商戶聯絡，在此特向已落稿商戶，深表歉意，並向下列公司銘謝：

通用無綫電行
飛龍電子公司
南記電子公司
構思實業公司
鴻圖電子公司
義興電子公司
宇宙電腦學校
萬事利電子公司
金鋒電子零件行
金獅電子貿易公司
金駛電腦系統公司
高星電子零件有限公司
金寶麟電子儀器有限公司
CPU ELECTRONIC CO.
RINGO ELECTRONICS CO.
MIRCOTECH ELECTRONICS CO.

如有令商戶帶來不便之處，敬請原諒。

電子科學出版社
INTERFACE ELECTRONICS
6502

STAMP
HERE

九龍新蒲崗衍慶街
衍慶大厦六樓B3室

訂閱通知書

茲訂閱 INTERFACE ELECTRONICS 6502 月刊 份 訂期：由第 期起

寄法：☐平郵 ☐空郵 ☐掛號

☐附上劃綫支票 No. ______

＊＊支票收款人請寫 "INTERFACE ELECTRONICS PUBLISHERS"

訂戶姓名______ 電話______

地址______

（外埠訂戶請用英文大楷）______

全年十二期訂閱費

香港 HK $72.00　　空郵東南亞地區 HK $190.00　　（掛號另加HK $48）

外埠 HK $140.00　　其他地區 HK $240.00

微型電腦袖珍辭典



Absolute loader A loader which will load into the computer memory absolute addresses.

Absolute address The actual address of a unit of information in store expressed in machine code.

Accessories Items which enhance the utility of a possession. They help satisfy the urge to go out and buy something new when you're not in the mood for a major purchase. See PERIPHERAL.

Access Time Time taken for an address to be sent to and data to be returned from memory using the longest path.

Accumulator— In a microprocessor, the internal register in which logical operations are performed and the results initially stored; characters may also be input to or output from the accumulator.

Acoustic Coupler A form of modern which permits attachment of an ordinary telephone handset so that a computer can communicate over any telephone. See MODEM.

Acronym A word made from the initial letters of a phrase. In the early days of computing many experts spent as much time forming clever acronyms as programming. They are a bit out of fashion today. See BASIC.

A-D— Conversion of continuous, analog data (like meter readings) into digital form that computers can read.

Add Arithmetic operation. Binary addition of two numbers involving the accumulator.

Address — The number used to refer to a specific byte in memory or to an input or output port.

Aim Avalanche induced migration. Semiconductor technology concerned with the making of memory links during the programming process.

Alphanumerics — Computer output or input in the form of letters and numbers rather than graphs or drawings.

ALU The arithmetic and logic unit that forms part of a central processing unit. It is the main routing element internal to the computer.

Aluminised (paper) Printer paper surfaced with a thin deposit of metal. The characters are formed by darkening caused by electric current flowing onto the paper from the pins of a matrix print head.

Analog Where you are just before you fall off the log. Seriously, an analog device uses a physical quantity, such as length or voltage, to represent the value of a number. The now obsolete slide rule was an analog calculator, where length along a piece of wood or plastic represented numerical quantity.

And A logical operation in which the result is true if all inputs are true and false if at least one input is false.

Architecture The word applied to hardward and software configura-

Architecture The word applied to hardward and software configurations in the way that the parts fit and communicate together.

Array A list of numbers of strings (or other entities in more sophisticated systems), elements of which can be referred to by their position in the lsit.

ASCIA Asynchronous Communication Interface Adaptor. A peripheral chip which can control the transmission and reception of data to and froma serial asynchronous peripheral such as a visual display unit or teletype.

ASCII American Code for Information Interchange that consists of a character set and an 8 bit code representation.

Assemble To translate from a symbolic program to a binary (machine code) program.

Assembler Software program which translates instructions ir assembly code (symbolic operation code) to binary machine language.

Asynchronous The ability to wait until told to 'go'.

BASIC Beginners' All-purpose Symbolic Instruction Code — a popular high-level programming language developed at Dartmouth College, USA.

BASIC (Integer) Concerned only with whole numbers, cutting off any fractions or decimal parts.

Baudrate The rate of data transmission in serial communications links.

Batch (Processing) A method of computer working in which a large number of transactions are grouped together before processing (so that control totals etc can be taken) and which are then passed through the various stages of processing as a group or batch. This was the original method of data processing for commercial work and contrasts with *interactive* and demand processing.

BCD Binary Coded Decimal — a 4-bit system for representing the 10 decimal digits.

Benchmark A standard computing task used to measure the relative speeds of different processors.

Binary Numbering system with the base 2, using the digits 0 and 1 instead of the decimal series 0 to 9. All digital computers work on data and instructions presented as binary numbers.

Bit Binary digit (contraction). Must be 0 or 1.

Block A sequence of **data words** or **bytes** treated as a unit, especially when working with magnetic tape.

Boot An instruction or very short program which will initiate a computer's operating system (short for bootstrap).

BPS Bits Per Second — a rate of data transmission between devices Eg 300 bps is a popular rate for some terminals, roughly equivalent to 30 characters per second (cps or chps)

Branch Normally, a computer executes the statements of a program in order of appearance. Statements that tell the computer to break out of this normal mode are said to cause a branch. In BASIC, one such statement is a GOTO.

Breakpoint Program address at which execution is halted to allow debugging or data entry.

Bubble memory A compact, high-capacity random access *memory* device which holds *data* as minute magnetic domains or 'bubbles' The data is not lost when power is removed.

Buffer (1) An area of memory designated to hold *data* being transferred between d

Buffer (1) An area of memory designated to hold *data* being transferred between devices working at different speeds, eg the fast processor and the slower keyboard, printer or disc.
(2) An electronic device in a signal path designed to allow signals to pass in one direction but to hold back unwanted reverse voltages which might damage the sending apparatus.

Buffer Store A temporary memory store between the central processor and input/output peripherals.

Bug A fault in the computer or mistake in the program.

Bus (or Buss) The means used to transfer information from one part of a computer to another.

Bus Structure — A fixed arrangement of the wires of a bus.

Byte The most commonly used byte size consists of 8 bits, hence a 16 bit word could be assembled from 2 bytes.

微型電腦袖珍辭典



CAD/I/L Computer Aided Design/Instruction/Learning.

Cartridge A protective carrier of magnetic tape (a variant of the familiar cassette) or *disc.*

Central Processing Unit Usually referred to as the "CPU." See CPU.

Chain A process whereby one computer program automatically follows another.

Character Set A set which can include alphabetic letters, numerals, punctuation marks, mathematical operations etc.

Chip 1. A small (typically less than half a centimeter on a side and quite thin) piece of material (usually silicon) into which have been formed from a few dozen to tens of thousands of circuit elements. This is done by etching the material, depositing microscopic metal conductors, and selectively impregnating ("doping") the material with various elements that change its properties. See INTEGRATED CIRCUIT. 2. The integrated circuit that houses a chip. This is a somewhat colloquial usage and should be avoided.

Clear To erase the contents in store.

Clock Time reference for a microprocessor.

Cobol Language for commercial uses and written using English type statements.

Code A system of symbols which have meaning to a computer language interpreter.

Command A request to the computer that is executed as soon as it has been received. Sometimes this word is used interchangeably with the terms "instruction" and "statement." Those terms properly refer to portions of programs and not to commands which are carried out immediately. See INSTRUCTION.

Compiler A program that translates one computer language into another. Most commonly the term refers to a program that translates a higher level language into the computer's native language.

Computer Any device that can receive and then follow instructions to manipulate information. In any computer, both the set of instructions and the information on which the instructions operate may be varied from one moment to another. A device whose instructions may not be changed is not a computer. The distinction between a programmable calculator and a computer is that the computer can manipulate text as well as numbers, whereas the calculator can only handle numbers.

Concatenation In computerese, the process of joining two strings to make a longer string.

Control Characters Characters or commands obtained by holding down the key marked "CTRL" while pressing another key on a keyboard.

Cursor A symbol placed on the screen to let you know where the next character you type will appear.

CUTS Computer Users' Tape System – a standard for recording data on cassette tape.

CP/M Control Program/Microprocessor. A popular disc-based operating system for microcomputers using the 8080 and Z80 processors.

CPS Cha acters (rarely cycles) Per Second (sometimes chps).

CPU The abbreviation of "Central Processing Unit," an absolescent term for that portion of the computer which controls peripherals and memory. The CPU was once a separate part of a computer but the term has lost its usefulness in personal computers where it refers to a tiny portion of one of the chips in the machine.

CRT – Cathode Ray Tube, a TV-type screen which may be used by a computer to display its alphanumeric and graphic output.

D-A – Conversion of digital data to be continuous, analog form; a circuit to perform this conversion.

Daisy wheel The typehead component of sequential printer – like the *"golf ball"* but faster – whose characters are held on the periphery of a serrated plastic disc.

Data – The information in a computer program that the computer what process must be done.

Data base A system for organising the elements of information in a machine-readable file so that a program can readily select from this data base any particular abstraction or combination of information that may be called for. For instance, a customer data base might include full details of all customers (as required for service and distribution departments as well as sales and marketing) and also of every service call and delivery as well as each item invoiced to these customers during a year or longer. A suitable program could access that data base to answer such questions as "identify the customers buying more than £1,000 of item in less than five deliveries and receiving less than two service calls in the year."

Data Bus A group of parallel connections used to carry instructions and data to and from the MPU chip.

Data Store Any medium for storing data in large amounts.

Debug To fault-find an erroneous program.

Dedicated This means that a device or system is exclusively used for one application.

Digital Used in computerese to describe information that can be represented by a collection of bits. See BIT. Most modern computers store information in digital form.

Dedicated This means that a device or system is exclusively sued for one application.

Digital Used in computerese to describe information that can be

Dip Acronym for "Dual In-line Package," the most common physical form for an IC which has two rows of leads that look somewhat like the legs of a caterpillar. See INTEGRATED CIRCUIT.

Direct Addressing An addressing mode where the address of the operand is contained in the instruction.

Direct Memory Access The ability to allow an input/output to control the central processing unit for memory recycling.

Disassembler A program that translates a computer's native language into assembly language.

Disc (Disk) Magnetic storage device allowing fast random access to any selection from a large volume of *data.* A fullsize hard disc will hold say 5 megabytes or more, a smaller *floppy* disc typically holds from 80 to 250 kilobytes but in either case the capacity is being increased all the time.

Display – A device that shows the computer output or status visually instead of on paper. The most frequently used displays are CRTs, or multi-segment LEDs (like calculators).

Document noun: A written description of a piece of software or hardward.
verb: To produce such a description.
Production of good documentation is a sure sign that a programmer is not a hacker. See HACKER.

Dos Acronym for "Disk Operating System" – a collection of programs which facilitate use of a disk drive. It is pronounced "de-oh-ess" or "doss."

Duplex A mode of data transmission where each station can send and receive simultaneously.

Dynamic (memory) Random Access Memory (RAM) requiring constant refresh signals but normally using less electrical power than *static* memory.

微型電腦袖珍辭典



EAROM Electrically Alterable Read Only Memory. Typically taking 10 msec to erase and 1 msec to write, this non-volatile storage might better be considered as 'Read Mostly Memory' as the write capability is likely to be limited to say 100,000 cycles.

Editing Making corrections or changes in a program or data.

English The proper language to be used between English speaking people to describe computers and how they work. See COMPUTERESE.

Eprom — Erasable PROM, a type of PROM which can be erased with ultraviolet light and then re-programmed.

Execute To do what a command or program specifies. To run a program or portion of a program.

Execution Time The time taken to perform an instruction in terms of clock cycles.

Expression A combination of variables, numbers and operators that can be evaluated to a single quantity. For example, if R has the value 8 then the numerical expression 4 * (R – 3) evaluates to 20. There are also expressions involving quantities other than numbers (e.g. strings) in some computer languages.

Feature A bug as described by the marketing department. See BUG.

Fetch The act of obtaining data from an addressed source such as memory or an input post.

Firmware Data or instructions stored permanently in read only memory.

Flag A binary digit which can take values of 0 or 1 depending upon the condition that the flag is said to represent.

Flat Lines Inputs to a microprocessor controlled by input/output devices. Decision making instructions called 'jump instructions' are used to test these.

Floppy Disk Mass storage method which uses flexible material similar to magnetic tape.

Floppy (disc) A mass-storage device comprising a soft (floppy) plastic disc with magnetisable surface on which data is recorded and may be accessed rapidly by a moving read/write head. The disc, either 8" or 5¼" in diameter, rotates inside a protective cardboard sleeve.

Format verb: To specify the form in which something is to appear. noun: Such a specification.

Fortran — FORulat TRANslator, a highlevel language designed for scientific programming.

Graphics — The processing, input and output of data other than alphanumerics or control or status functins (for example, pictures, images and graphs).

Golfball A type of typewriter (or the print head from which it gets its name) in which the print characters are embossed on the surface of a sphere very similar in size to a golfball. Rotation of the sphere brings the appropriate character into line for each required impression. The process is usually slow (15 cps) but of good quality.

Hacker A person who likes to play with computers, usually to the exclusion of all other activities. These creatures are often asocial, and their greatest joy is creating a program which makes a fellow hacker ask "how the hell did you get the computer to do that?"

Handshake System of data transfer between central processor and peripheral conditional to peripheral's agreement to accept.

Hard Copy System output that is printed on paper.

Hardware The mechanical and electronic components comprising a system.

Hex/Hexademcimal — A number system with base 16. The sixteen digits are: 1, 2, 3, 4, 5, 6, 7, 8, 9, O, A, B, C, D, E, and F.

High-level language — A computer language that humans can understand easily and that a computer can translate into the machine-language form it understands, in one or more steps. Many instructions in a high-level language require that the computer perform more than one computer operation. Some of the more common high-level languages are : BASIC, COBOL, FORTRAN, and PASCAL.

IBM Card See PUNCHED CARD.

IBM Machine Obsolescent term for "computer." It properly refers to anything made by International Business Machines, but the term used to be a generic appellation for computers since IBM used to make such a large percentage of them.

IC See INTEGRATED CIRCUIT.

I—EEE Institute of Electronic and Electrical Engineers (in USA) — a body which has set a number of standards for more orderly interchange of informatin between various electronic devices, including computers.

Immediate Addressing An addressing mode in which the operand is located with the same instruction.

Impact (printer) One which forms characters by striking a ribbon onto paper and can therefore produce carbon copies.

Increment To increase the contents of a register of memory location by a fixed amount.

Indexed addressing A form of indirect addressing which uses a special location known as the index register to hold the address of the operand. Index registers can be incremented or decremented under program control.

Indirect addressing An addressing mode where the address of the location of the operand is contained in the instruction.

I/O — Input/output. A) The equipment used to put information into or take information out of a computer. B) The information itself that is given to or taken from a computer.

Initialize 1. To set up the starting conditions necessary for the execution of the remainder of a program. For example, in a program that draws a circle, the initialization might include specifying the radius of the circle. 2. To prepare a diskette so that the computer can later store data on it.

Instructions Binary words; each particular combination of bits representing a particular command to open gates, generate shift pulses and increment counters. Typical examples are SHIFT, STORE, ADD etc.

Instruction decoder Decorder used for interpreting instructions fetched from memory.

Instruction register The register within a microprocessor in which the address of the circuit instruction is stored.

Instruction set The set of instructions that a particular microprocessor is able to perform.

Integrated Circuit Usually consists of a plastic or ceramic body from 1 to 5 cm in length, from 1 to 2 cm in width, and typically 2 or 3 mm thick with from 4 to 40 (rarely more) metal leads extending from it. It is usually called an "IC" (pronounced eye-see). Inside the body is a chip (SEE CHIP). The body protects the chip, and the leads allow electrical connection of the chip to other components. The term "chip" is often incorrectly used to refer to the entire IC.

Intelligent terminal An input/output device which includes its own logic circuits and memory so that, for instance, data may be validated or changed in format before transmission to the main computer.

Interactive A working arrangement under which the computer reacts immediately to respond to any mistakes which may be made by the user or to reply to his enquiries as soon as they are expressed. In some business activities, as also in program writing, this leads to much faster progress than would otherwise be possible.

Interation See LOOP.

Interface The interconnection arrangements between a computer and devices, such as printer or modern, attached thereto.

Interpreter A program to translate a high level language (typically BASIC) to machine language and to execute each instruction, line by line, immediately.

Interrupt A signal which suspends processing to allow some other command to be obeyed.

Interrupt line An asynchronous input to the microprocessor chip which when asserted causes the MPU to enter the interrupt state. Some microprocessor systems have a number of interrupt lines, and each of these is assigned a priority so that the one with the highest priority is serviced first.

Interrupt vector This is the address at which the start of an 'Interrupt Handler' program will be found. Some microprocessors used fixed interrupt vectors set by the chip manufacturers, while more sophisticated systems with many possible interrupt sources allow the interrupting device to provide its own vector via an input port.

Iterate To repeatedly execute a set of instructions.

Jump Unlike one-stop incrementing this is an instruction which causes the program to leap backwards or forwards, either by a specified amount or to a specified location.

"K"/Kilobyte— 2^{10} (1024) bytes. Memory is usually reckoned in kilobytes. "K" stands for "binary thousand" (1024), while ordinary decimal thousands are represented by lowercase "k".

Kansas (City) A standard for recording programs and data on cassette tape named after the city where a conference was held at which the standards were agreed.

Keyboard A group of keys. Among those who work with computers, it is usually used to mean a typewriter-like layout of keys (for numbers, letters, punctuation and other symbols) plus the accompanying electronics, that is used to input information into a computer. Other arrangements of keys are usually referred to as keypads.

Keyword or Key Word A word that has meaning in a computer language. See RESERVED WORD.

Label A string of characters used instead of absolute addresses in computer programs. These are usually alphanumeric.

Language Means of communicating data and in structions to a computer.

Library A collection of standard routines, or programs, accessible to the computer user.

Lifo Last in first out — term describing data stack.

Light pen A stylus with a light sensor which allows a computer to identify the point at which a Video Display Unit (VDU) screen is being touched.

Line printer A computer peripheral which prints a whole line at a stroke, instead of doing each character sequentially.

Link Jump instruction that enables return from a sub-routine to the main program.

Load To copy a program (eg from tape or disc) into **memory**, leady for execution.

Location A group of storage elements in a computer memory which areidentified by a numerical address and able to store a computer word.

Loop See ITERATION.

Love An infantile human disorder to which computers are as yet immune.

LSI Large Scale Integration — the combination of more than 100 logical gates (or decision switches) in a small silicon chin

Machine Code A low level language consisting of the strings of binary digits (0's and 1's) that the processor can interpret and execute.

Mainframe A relatively large computer distinguished from the peripherals which, with the mainframe, complete the configuration. The term derives from times before integrated circuits, when processors were wired up with large numbers of separate components mounted on circuit cards or boards which were in turn mounted in metal racks or frames enclosed in one or more large metal cabinets.

Mass Storage — Recording systems of holding or storing programs or data not required for immediate use. Such information must be read into the computer before it can be used. Common forms of mass storage used with home computers are cassettes and floppy disks.

Matrix (printer) A printer whose characters are formed by selecting a pattern of dots from a matrix typically 5 dots wide and 7 high.

Memory Immediate access data storage, directly addressable by a central processor and typically comprising a combination of RAM and ROM chips.

Menu A list of options from which to choose.

Micro — (also μ) Prefix signifying one millionth. Also used descriptively of something very small, though not as small as nano— or pico—

Microcomputer A computing system using a microprocessor.

Microfiche A guppy.

Microprocessor An integrated circuit that performs the task of executing instructions. The presence of a microprocessor in a product does not make it into a computer.

Microprogram The program, usually stored within the micro-processor, which controls it during the basic fetch/execute sequence.

Minicomputer A somewhat vegue term for the middle range of computers. Machines addressing up to 64K bytes or words of memory tend (at the present time) to be called Microcomputers and machines able to address more than 64K memory locations tend to be called Minicomputers unless they separate into distinct parts, in which case the processor part may be called a Mainframe.

Mini-floppy The smaller size of floppy disc, 5¼" in diameter.

Mnemonic An easily remembered substitute for a word or phrase.

Modern Acronym for MOdulator/DEModulator — a device adapting computer data for transmission by telephone line and vice versa.

Modulator A device that lets a computer use any ordinary television set for output. This term is understood in this sense mainly with respect to personal computers as such modulators are not generally used with larger machines. It is sometimes referred to as an RF modulators are not generally used with larger machines. It is sometimes referred to as an RF modulator. RF stands for "Radio Frequency," meaning Television Broadcasting Frequency. Such is life.

Monitor The first level of computer operating systems: the program which turns machine code commands into action, managing input, output etc.

n-sec Nanosecond, one-thousand-millionth of a second.

Native Lanugage The language that a computer was built to understand. This lanugage is usually rather inconvenient to use. Thus, most computers are provided with other languages as well.

Non-Volatile— Memory that retains its contents even if no power is supplied to it. (See RAM.)

Null String A string consisting of no characters whatever. If it doesn't seem like a useful idea, think about how useful the number zero is.



Octal — A number system with base 8. The eight digits are: 0, 1, 2, 3, 4, 5, 6, and 7.

Operand Data used in a machine operation. The item of information that is manipulated or used in an instruction.

Operator 1. A symbol that represents a mathematical operation, such as addition, divisiion, comparison or exponentiation, to name a few. 2. In the old days, a specialist who knew how to operate a computer. They usually wore white lab coats.

Operation Code That part of a machine-code instruction that specifies the operation to be performed by the instruction.

Operating System The computer's resident program which determines how instructions, input and output devices etc are managed.

OS An acronym for "Operating System" (pronounced "Oh Ess"). An OS is a collection of programs which allege to aid a person in controlling a computer. This term is usually used in reference to large computers. A small computer operating system is often called a monitor. See MONITOR.

Output Information leaving from a computer can be displayed by a printer or CRT. This term can also be used as a verb, even if it does sound a bit awkward, as in: Watch the computer output a graph. See INPUT for a more elaborate discussion.

Page The largest area of memory that can be addressed by the computer's addressing scheme.

Package A set of programs designed to perform a common task, eg payroll, generalised to suit a variety of users. A turnkey package may comprise both the programs and the equipment on which they run.

Paper Tape — A mass storage system using paper tape with eight hole-positions representing the eight bits of each byte stored.

Parallel — A type of I/O in which each bit of a computer word is transmitted over a separate wire simultaneously. (See serial.) A computer's internal information flow is also normally in parallel form.

Pascal — A comparatively new computer language, now becoming available for home computers.

Patch A small piece of computer program inserted in a longer program to remedy some bug or defect in it.

Peripheral Device attached to a central processor, eg printer, plotter, disc unit, but not necessarily essential to its use.

Personal Computer A general purpose computer that is inexpensive enough to be owned by an individual with a moderate income.

Pilot A programming language for small computers, designed to be particularly appropriate for teaching in schools.

Plotter Computer-driven graphical display using pen on paper.

PORT — The I/O circuit that connects a computer with a peripheral. Each port has an address (number) by which the computer can distinguish it from other ports. Ports may be serial or parallel.

Pointer The register containing the address of the memory location to be accessed.

Precedence Rules that state which operators get executed first in an expression. For example, by the usual rules of precedence 3 + 4*2 is 11 rather than 14 since multiplication has "higher" precedence than addition and is done first.

Prime Number The most tender and most juicy-kind of number, as certified by the U.S. Department of Agriculture.

Printer In computerese: a peripheral that makes hard copy of letters and numerals. A line printer prints a whole line of text at a time. A serial printer prints one character at a time.

Processor — A set of circuits capable of performing the essential functions of a computer CPU. In the case of a microprocessor (MPU), these circuits are combined into one or a few integrated circuits.

Program A sequence of instructions that describe a process. A program must be in a language that a computer can understand.

Program Counter The register which contains the address of the next instruction to be executed.

Programmer A person who writes and documents programs.

PROF — Programmable ROM, non-volatile memory which can be programmed by a user. Two subdivisions of this class are EPROM (Erasable FROM) and EAROM (Electrically Alterable ROM) which are erasable with ultraviolet light or electrical impulses respectively. EPROM and EAROM may be re-programmed after they are erased.

Prom Programmer A hardware facility for storing data or programs into PROM memory.

Prompt A symbol that appears on your computer's display to let you know that it is ready to pay attention to your commands.

Punched Card An obsolescent means of storing information by punching holes in a small sheet of cardboard. Creates litter

RAM Acronym for "Random Access Memory." This is the main memory of any computer. Information and programs are stored in RAM, and they may be retrieved or changed by a program. For some computers, the information in RAM is lost whenever the power is turned off.

Random Number Generator A program that, when executed, provides a number whose value is difficult to predict. In many computer systems a random number generator is built into the language rather than appearing as a subroutine.

Reserved Word A word that you cannot use as a variable name, since it has been pre-empted for use in the computer's language. You also may be restricted from using reserved words in other ways as well. Key words are often reserved words. See KEY WORD.

Reset (button) A switch whereby computer control is returned to the monitor or low-level operating system and all internal variable values are changed to zero. This may be the only way of getting out of some endless loop which has arisen from a programming error.

Return The key and corresponding computer instruction which sends the contents of keyboard buffer into a computer's memory for execution (term derives from 'carriage return' on a typewriter).

ROM Read Only Memory.

R-F Modulator — Radio-Frequency Modulator, a device that converts video information into frequencies acceptable to a TV set so that the TV may be used as a substitute for a video monitor.

RUN The instruction to execute a program.

Save To store a program somewhere other than in the computer's memory, for example on a diskette or cassette tape.

Scratch Pad Short access time memory for storing data.

Screen 1. A surface on which information is displayed, such as a TV screen. 2. The sound you make when a program refuses to work for the seventeenth time.

Scroll To move all the text on the screen (usually upwards) to make room for more text (usually on the bottom).

Semantics In computerese: the meaning of an instruction or program, i.e. what it is supposed to do.

微型電腦袖珍辭典



Serial Things occurring one after the other. A serial interface is one that sends out one bit at a time. See PRINTER for another use of the term.

S-100 Name of a bus or connection standard shared by many manufacturers and employing 100 connection positions. Unfortunately, there are some minor variations between different manufacturers' versions of the S–100 bus but the I-EEE has now defined a universal standard for it. Primarily designed as a *memory* bus and not for general purpose use.

Software The different kinds of program required to work a computer.

Source code A program written in one of the high-level languages and requiring compilation into machine language before use.

Special Character A character than can be displayed by the computer, but is not a letter or numeral. Here are some special characters:
^ ~! ¨ # $ % & ' ( ) – = [ ] \ ' @ ¨ ; + : * , < . > / ?

Stack Area of store used for temporary storage of data.

Statement An instruction.

Static RAM Random Access Memory which does not require continuous refresh signals but tends to use more power than Dynamic RAM and still loses its contents when power is removed.

Status Register This contains information on the internal status of the computer.

String A sequence of alphanumeric characters.

Subroutine A rountine within a complete program which can be entered from anywhere and from which control is passed to the next sequential instruction in the main program. Subroutines make for efficiency as they only need to be written once and jumped into only when required in the program.

Syntax The grammar of a programming language.

System Routine A routine or subroutine, held in this monitor library, not included in but required by a program.

Teletype A noisy, slow and obsolescent terminal that may include facilities to handle paper tape.

Terminal The condition of a computer just before it dies. Actually: A peripheral, consisting of a keyboard through which a person can send information to the computer and a printer or screen through which the computer can present information to the person.

Text In computerese, this word means data other than numbers.

Thermal (Printer) A matrix printer wherein the print impression is made by heating a selected pattern of wires within a matrix (say 5 x 5 x 7) so that the heat causes points on the specially-treated paper to darken, to form the selected character.

Time-Sharing A method of operating a computer whereby two or more users apparently enjoy simultaneous access to and control of the machine. In practice what is happening is that the computer is attending to the users one at a time, but in a sequence of time intervals so short that none is normally aware of any delay.

Tristates Logic devices which have normal logic levels and a high impedance state which represents a gate disable (e.g. a tristated *data bus represents a disconnection*).

UART Universal Asynchronous Receiver/Transmitter. UART is an alternative and more popular name for the ASCIA.

USART Universal Synchronous/Asynchronous Receiver/Transmitter. An improvement on the UART in that it may be programmed to operate as a synchronous communications link.

Variable A name for a quantity. A variable in a computer language can be thought of as a box into which a value may be stored. Such values are, typically, numbers and strings. More sophisticated systems may allow the value of a variable to be a picture, an array or some other structure.

User The person who programs and operates the computer.

Users' Group An association of people who all have an interest in a particular computer or group of computers. They usually meet to exchange information, share programs, trade equipment, and show off their accomplishments.

VDU Visual Display Unit. This displays data in the form of characters on a T.V. type screen.

Vector Memory address provided to the processor to direct it to a nece area of memory.

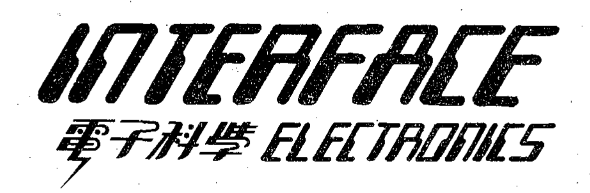
Volatile Memory Describes a RAM where data is lost when the power applied is removed.

Window A portion of the computer's display that is dedicated to some special purpose.

Word The specified number of *bits* that a computer is organised to process as a group — eg 16-bit word: but the popular 8-bit word is called a byte.

Word Processor A computer with software for entering, editing, storing, formatting and printing text, rather than processing figures.

Write Alternatively, to output information or to store data in memory.



用 VIC 20 砌大型字母

```
10 DIMB$(45,4),C$(45):L=45
20 FORI=1TOL:READC$(I):FORJ=1TO4:READB$(I,J):NEXTJ,I
30 U$="[illegible]"
90 PRINT"[illegible]";
100 GETG$:IFG$=""THENGOSUB300
105 IFG$=CHR$(13)THENT=0:GOTO90
106 IFG$=CHR$(20)THENF=1:GOSUB300:T=T-4:IFT<1THENT=0:GOTO115
110 GOSUB200:IFX=0THENG$="":GOTO100
115 GOSUB220
120 PRINTU$;TAB(T);
140 IFT>18THENPRINT"[illegible]":T=0
150 GOTO100
200 X=1:FORJ=1TOL
210 IFC$(J)=G$THENC=J:RETURN
215 NEXTJ:X=0:RETURN
220 FORI=1TO4
230 PRINTTAB(T)B$(C,I):NEXT
240 IFQ=0THENT=T+4
290 Q=0:RETURN
300 IFF=1THEN350
310 G$="CR":F=1:Q=1:RETURN
350 G$=" ":F=0:Q=1:RETURN
500 END
1000 DATA"A",[illegible]
1010 DATA"B",[illegible]
1020 DATA"C",[illegible]
1030 DATA"D",[illegible]
1040 DATA"E",[illegible]
1050 DATA"F",[illegible]
1060 DATA"G",[illegible]
1070 DATA"H",[illegible]
1080 DATA"I",[illegible]
1090 DATA"J",[illegible]
1100 DATA"K",[illegible]
1110 DATA"L",[illegible]
1120 DATA"M",[illegible]
1130 DATA"N",[illegible]
1140 DATA"O",[illegible]
1150 DATA"P",[illegible]
1160 DATA"Q",[illegible]
1170 DATA"R",[illegible]
1180 DATA"S",[illegible]
1190 DATA"T",[illegible]
1200 DATA"U",[illegible]
1210 DATA"V",[illegible]
1220 DATA"W",[illegible]
1230 DATA"X",[illegible]
1240 DATA"Y",[illegible]
1250 DATA"Z",[illegible]
1260 DATA"£",[illegible]
1270 DATA"!",[illegible]
1280 DATA"%",[illegible]
1290 DATA"&",[illegible]
1300 DATA"'",[illegible]
1310 DATA"*",[illegible]
1320 DATA" ",[illegible]
1330 DATA".",[illegible]
1340 DATA"0",[illegible]
1350 DATA"1",[illegible]
1360 DATA"2",[illegible]
1370 DATA"3",[illegible]
1380 DATA"4",[illegible]
1390 DATA"5",[illegible]
1400 DATA"6",[illegible]
1410 DATA"7",[illegible]
1420 DATA"8",[illegible]
1430 DATA"9",[illegible]
1440 DATA"CR",[illegible]
```

太　空　戰

火星人胆敢攻擊地球，發動星際大戰，讓你駕駛ＶＩＣ－２０太空戰機，進行報復行動！

「太空大戰」是ＶＩＣ－２０的另一種電腦遊戲，以撲克牌中的梅花（ｃｌｕｂ）形狀出現的外星人，從螢幕的底部排列，而你則在螢幕的上空，移動飛船和發炮，每擊中一個外星人，你便可獲得兩分。時間與積分顯示在底線的上面。從密集的外星人中會間斷地發射出飛彈向飛船還擊，而打中這些飛彈可獲五分；但無論如何，你必須使飛船儘快離開它的彈道——因爲這些飛彈是不能毀壞，而且一旦飛船被擊中，會使你損失二十分之多。

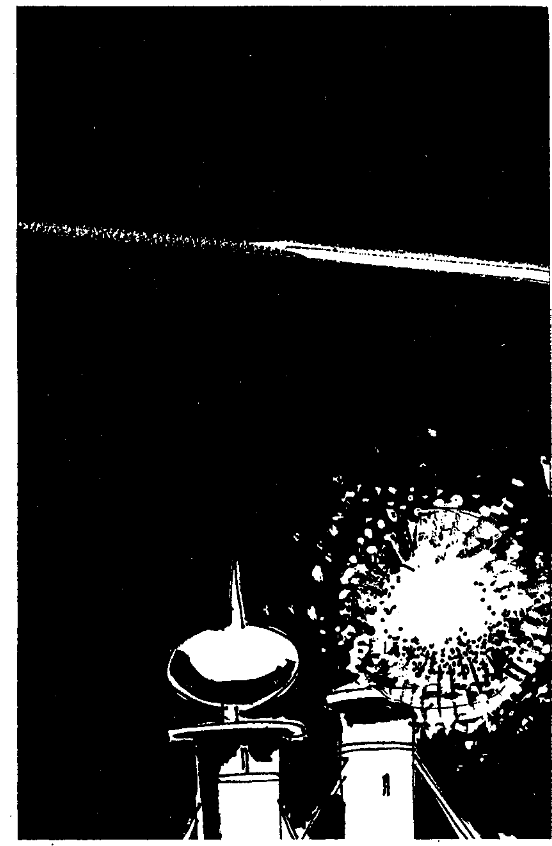
至於３Ｋ　ＲＡＭ擴充盒式磁帶的擁有者，則更可利用此機的高解像度能力，製造出更精細的外星人，飛彈的圖像。

加入３行４３０，減少３Ｍ值的循環至七次，以配合如下面的Ｐｏｋｅ指令：

```
430 FOR M = 1 to 7: POKE 36865,
  35+RND(1)*6 :NEXT M,T
```

３６８６４和３６８６５控制螢幕上Ｘ和Ｙ的位置。加上行４３０便可輕易地使畫面作上下的晃動。





```
READY.

 1 TI$="000000":POKE36878,15:POKE36879,8
 2 DIMA(22):GOSUB800
 3 Z1=7680+19*22
 5 PRINT"▮":Z=7680:PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮TTTTTTTTTTTTTTTTTTTTTT▮"
 6 FORI=1TO22:A(I)=1:POKE7680+22*22+I-1,121:NEXTI
 7 PRINT"▮▮▮▮▮HANG ON!"
 8 FORY=0TO505:POKE38400+Y,INT(RND(1)*7)+1:NEXTY
 9 POKE36879,8:PRINT"▮▮▮▮▮                    "
 10 G=PEEK(197):RT=0
 11 IFRND(1)<.3THEN19
 12 D=INT(RND(1)*22):RT=RT+1:IFRT=4THEN19:IFA(D)=0THEN12
 13 A(D)=0:R=R+1:IFR=15THENZ1=Z1-22:FORI=1TO12:A(I)=1:NEXTI:R=0
 14 POKEZ1+D,88
 19 IFTI$>"000130"THEN200
 20 IFG=29THENX=X-1
 21 POKE36877,0
 22 IFR/2=INT(R/2)THENPOKE36877,220
 23 IFL>0THEN300
 24 IFRND(1)>.9THEN300
 30 IFG=37THENX=X+1
 31 IFX<0THENX=0
 32 IFX>21THENX=21
 40 POKEZ+X1,32:X1=X
 50 POKEZ+X,81
 59 IFG=32THEN80
 60 PRINT"▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮▮                               ▮SCORE"SC"TIME "RIGHT$(TI$,
4)
 61 GOTO10
 80 FORI=1TO19
 81 POKE36876,244-I*4
 85 IFPEEK(Z+X+I*22)=88THENSC=SC+2
 86 IFPEEK(Z+X+I*22)=65THENSC=SC+5:POKE36879,0:FORP=1TO80:NEXTP:POKE36879,8
 90 POKEZ+X+I*22,93
 100 IFI>1THENPOKEZ+X+I*22-22,32
 110 NEXTI:POKE36876,0
 120 POKEZ+19*22+X,32:GOTO60
 200 POKE36877,0:FORR=1TO5:PRINT"▮▮▮▮▮▮   YOU SCORED"SC
 210 FORU=1TO80:NEXTU
```




```
220 PRINT"        YOU SCORED"SC
230 FORU=1TO80:NEXTU
240 NEXTR
250 PRINTCHR$(142):FORU=1TO100:NEXTU:POKE197,64:RUN
300 IFL=0THENQ=X
311 POKE36875,128+2*L
320 IFPEEK(Z+22*20-L*22+Q)=81THEN400
330 IFL>1THENPOKEZ+22*23-L*22+Q,32
340 POKEZ+22*19-L*22+Q,65
350 IFL>18THENL=0:POKEZ+Q,32:POKE36875,0:GOTO30
355 POKE36875,0
356 L=L+4
360 GOTO30
400 POKEZ+Q,42:POKE36877,220:POKE36879,0
410 FORT=15TO1STEP-1
420 POKE36878,T:POKE36879,8
430 FORM=1TO80:NEXTM,T:POKE36878,15:POKE36877,0:SC=SC-20:GOTO330
800 PRINT"             WANT INSTRUCTIONS?
810 GETA$:IFA$=""THEN810
820 IFA$="N"THENPRINTCHR$(142):GOTO3
825 PRINTCHR$(142)
830 IFA$<>"Y"THEN810
831 POKE36879,8
835 PRINT"   ALIENS WHICH ATTACKED  EARTH HAVE RETREATED              TO MARS.
"
840 PRINT"           YOUR TASK IS TO KEEP  THEM THERE UNTIL       FURTHER HELP A
RRIVES."
850 PRINT"   WATCH OUT FOR THE      DEVIOUS MISSILES!"
860 PRINT"            HIT ANY KEY "
870 GETA$
871 IFA$<>""THEN875
872 FD=FD+FT:IFFD>9THENFT=-1
873 IFFD<1THENFT=1
874 POKE7680+5*22+FD-FT,32:POKE7680+5*22+FD,88:GOTO0870
875 PRINTCHR$(142)
880 PRINT"   <-MOVES YOU LEFT       >-MOVES YOU RIGHT      'SPACE BAR -FIRE"
890 PRINT"           ' '-IS YOUR SHIP     HELP ARRIVES IN 90     SECONDS"
900 PRINT"          HIT ANY KEY "
910 GETA$:IFA$=""THEN910
920 POKE36879,8:GOTO3
```

打磚遊戲

VIC—20　　打磚遊戲

此遊戲程式乃「打磚Ｂｒｅａ－ｋｏｕｔ」另一簡單的版本，其中大部份使用Ｐｅｅｋｓ和Ｐｏｋ－ｅｓ指令。利用Ｚ鍵控制左移而Ｃ鍵用來控制右移。程式中的１５行至７０行，使電腦繪出多道不同顏色的磚牆。然後在畫面的底部，從一個「隨機」的位置出現一個光點，這個「波」在移動，反彈的過程中，不斷碰撞到畫面上的磚牆和把一塊一塊的磚打下。

而進行此遊戲者，則需利用出現在螢幕底部的「拍子」把每次反彈回來的「波」打回螢幕的上方，以期擊下另一塊磚頭。如果一旦落空，而使光點到達畫面的底部，四次機會中便會減少一次（即每局遊戲中，只能落空四次）。又如果，在一局中，能把所有磚頭打光，那這局進行遊戲者便算贏了。

筆者在進行這個程式的試驗時，發覺只要加入下列的行１３至４７。這程式便可改爲使用歡樂棒Ｊ－ｏｙｓｔｉｃｋｓ進行，這些額外的數行，即可令歡樂棒的推上使拍子左移，拉下使拍子右移。

```
 2 REM******BREAKOUT******
 4 REM      11/1/82
 5 PRINT"□■":POKE36879,9
 6 PRINT" THIS IS THE GAME OF"
 7 PRINT"      ▣BREAKOUT■":PRINT:PRINT"THE Z-KEY MOVES LEFT":PRINT"THE C-KEY MOVE
S RIGHT"
 8 PRINT:PRINT" HIT ANY KEY TO PLAY"
 9 GETA$:IFA$=""THEN9
 10 POKE36878,15:Y=20:DY=-1:C=4:GOSUB1000:PRINT"□"
 15 FORJ=2TO7
 20 FORI=0TO21
 30 POKE38444+I+(J-2)*22,J
 40 NEXTI:NEXTJ
 50 FORI=0TO131
 60 POKE7724+I,160
 70 NEXTI
 75 POKE7680+X+22*Y,81
 80 POKE8142+D,98
 90 POKE8143+D,98
 100 POKE8144+D,98
 110 S=PEEK(197)
 140 IFS=33THEND=D-1:POKE8145+D,32
 150 IFS=34THEND=D+1:POKE8141+D,32
 160 IFD<0THEND=0
 170 IFD>19THEND=19
 180 POKE7680+X+22*Y,32
 190 X=X+DX
 200 IFX<1ORX>20THENDX=-DX:POKE36876,220
 210 Y=Y+DY
 220 IFY=22THEN300
 225 IFY=0THEN400
 230 IFPEEK(7658+X+22*Y)=160ANDDY=-1THENDY=1:POKE36876,230:POKE7658+X+22*Y,32
 240 IFPEEK(7702+X+22*Y)=98THENDY=-1:POKE36876,240
 250 POKE36876,0
 260 GOTO75
 300 POKE36874,140
 310 FORI=0TO1000:NEXT
 320 POKE36874,0
 330 C=C-1
 340 PRINT"▣▣▣▣▣▣▣▣▣▣▣▣▣▣YOU HAVE ";C;"CHANCES            LEFT"
 343 FORT=0TO1500:NEXT
 345 FORK=0TO34:POKE7922+K,32:NEXT
 350 IFC>0THEN:Y=20:DY=-1:GOSUB1000:GOTO75
 355 GOTO460
 400 FORU=0TO5
 410 POKE36875,240
 420 FORI=1TO90:NEXTI
 430 POKE36875,0
 440 NEXT
 450 PRINT"▣  CONGRATULATIONS        YOU'VE DONE IT"
 455 PRINT"▣▣▣▣▣▣▣▣▣▣▣▣▣"
 460 POKE36878,0
 470 END
 1000 X=INT(RND(1)*20)+1
 1010 DX=INT(RND(1)*3)-1
 1020 IFDX=0THEN1010
 1030 RETURN
READY.
```

RINGO ELECTRONIC COMPANY

Shop No. 34, 1/F., Golden Shopping Centre, Fuk Wah St., Sham Shui Po, Kowloon.

深水埗欽州街高登商場34號

3-703500

RINGO ELECTRONICS CO.

致意

INTERFACE
電子科學 ELECTRONICS

MICROTECH ELECTRONICS

SHOP NO. 38, GOLDEN SHOPPING CENTRE,
SHAM SHUI PO, KOWLOON.
TEL: 3-874428
深水埗欽州街高登商場38號

MIRCOTECH ELECTRONICS CO.

致 意




INTERFACE
電子科學 ELECTRONICS


INTERFACE
電子科學 ELECTRONICS

金駿電腦系統公司

CAMELOT COMPUTER SYSTEM

SHOP NO: 5–6 1/F., GOLDEN SHOPPING CENTRE,
YAM CHAU STREET, SHAM SHUI PO, KOWLOON, H.K.
TELEPHONE NO: K-788474

金駿電腦系統公司

致意

INTERFACE
電子科學 ELECTRONICS

---

cpu ELECTRONICS. ENG. CO.

Rm. 12, 2/F., Fu Tao Bldg.,
98 Argyle St.,
Mongkok, Kowloon, H. K.
Tel: 3-7132345

九龍亞皆老街九十八號
富都大厦叁樓12室
電 話：3-7132345

CPU ELECTRONIC CO.

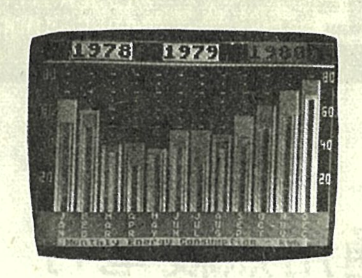
致意

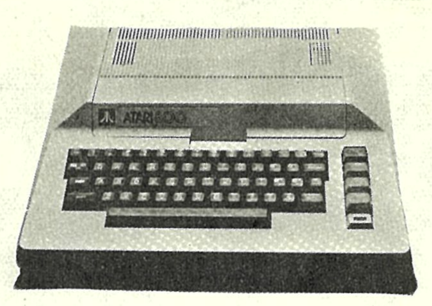


特別預告：

ATARI

勁





BAN SZETO

下期推出最新
最具爆炸性遊戲

保證 勁 GAMES

超級內容

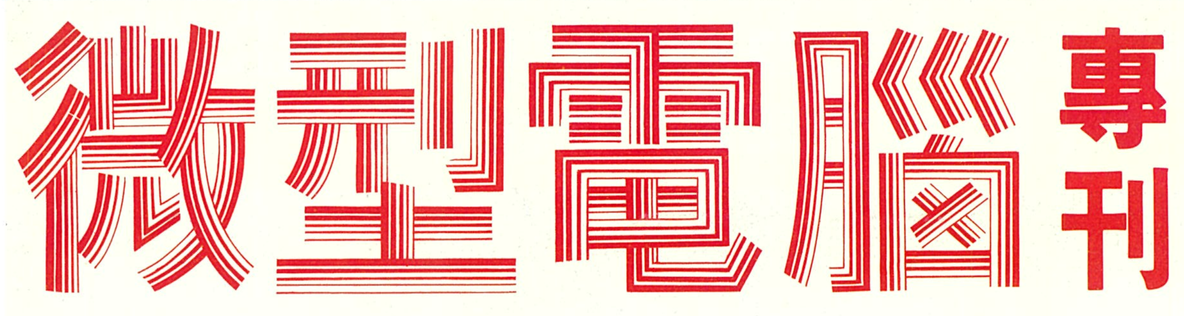
APPLE

布老圖

傾力泡製各類APPLE GAME

將全部公開

與你共享MICRO的樂趣




MICRO SPECIAL MONTHLY

專題：軟件保護淺談

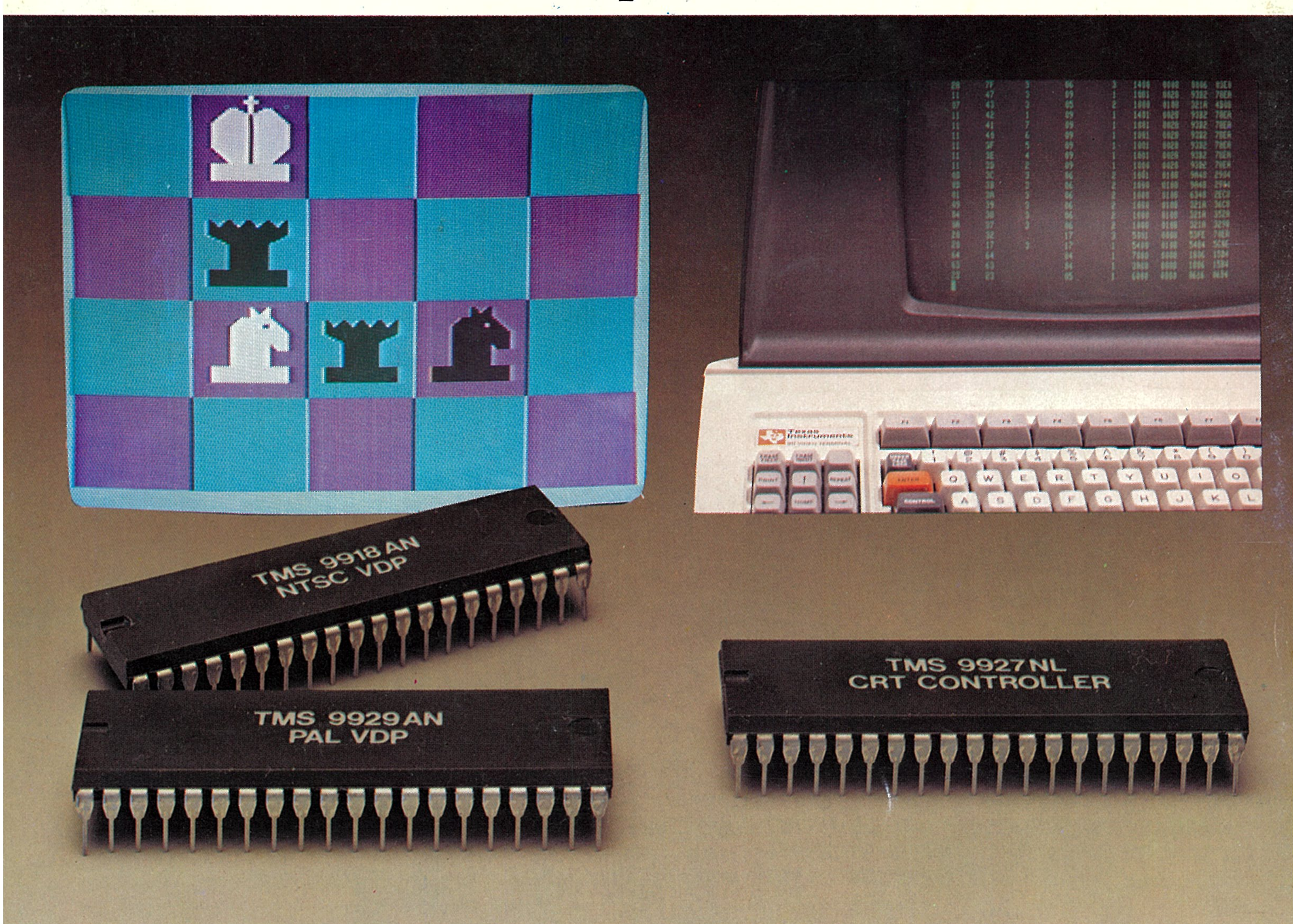
太空追擊戰

PC-8001 文字處理器程式

MZ80A 更詳細的介紹

PC-1500 利用拓撲原理的立體打井遊戲

TI introduces high design flexibility at low system cost.



TMS9918A NTSC Video Display Processor
TMS9929A PAL Video Display Processor

Key Features • Single-chip interface to color TV's (excluding RAM and RF modulator). • 256 × 192 resolution on TV screen • 15 unique colors plus transparent • 2 color graphic modes — complex color and pattern display • 40 character × 24 line textual display • General 8-bit bidirectional interface to CPU • Direct wiring to 4K, 8K or 16K dynamic RAM memories • Automatic and transparent refresh or dynamic RAMs • External video input capability • NTSC/PAL standard composite video output • Unique planar representation for 3D simulation • Single 5-volt operation • Standard 40-pin package
Typical Applications • Color computer terminals • Home computers • Drafting/design aids • Teaching aids • Industrial process monitoring • Home educational systems • Animation aids • Video games

TMS9927 CRT Controller

Key Features • Standard and Nonstandard CRT Monitors • Interlaced or Noninterlaced Formats • Scrolling Capability • Programmable Display Format: Characters per row Rows per frame Raster scans per row Raster scans per frame • Programmable Monitor Timing: Blanking Horizontal Sync Vertical Sync Composite Sync • Programmable via Microprocessors or PROMs • Cursor Output • Light pen application is allowed • TTL/NMOS compatible • Any CPU/MPU compatible • N-channel, Silicon Gate MOS Device • Standard 40-Pin Plastic or Ceramic Package
Typical applications • Intelligent Terminal • Smart Terminal • Information Display • Video Games

For further information, contact:
Texas Instruments Asia Limited Semiconductor Components Group
Rm. 902, Asian House, 1, Hennessy Road, Hong Kong.
Tel. 5-279041 Telex: 73809 ASIAT HX

Texas Instruments – where electronic innovation comes first.

TEXAS INSTRUMENTS
INCORPORATED